

STUDIES AND ANALYSES OF THE SPACE SHUTTLE MAIN ENGINE

**Technical Report
On**

**FAILURE INFORMATION PROPAGATION MODEL
DATA BASE AND SOFTWARE**

Contract No. NASw-3737

BCD-SSME-TR-87-2

December 15, 1987

A. E. Tischer

Prepared For

**National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812**

BATTELLE
Columbus Division
505 King Avenue
Columbus, Ohio 43201-2693

N89-20158

(NASA-CR-183586) STUDIES AND ANALYSES OF
THE SPACE SHUTTLE MAIN ENGINE. FAILURE
INFORMATION PROPAGATION MODEL DATA BASE AND
SOFTWARE (Batelle Columbus Labs.) 571 P
CSCI 21H G3/20

Unclas
C197302

STUDIES AND ANALYSES OF THE SPACE SHUTTLE MAIN ENGINE

**Technical Report
On**

**FAILURE INFORMATION PROPAGATION MODEL
DATA BASE AND SOFTWARE**

Contract No. NASw-3737

BCD-SSME-TR-87-2

December 15, 1987

A. E. Tischer

Prepared For

**National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812**



**A. E. Tischer
Manager
SSME Study**



**N. H. Fischer
Manager
Space Systems Section**

BATTELLE
Columbus Division
505 King Avenue
Columbus, Ohio 43201-2693

ABSTRACT

The failure information propagation model (FIPM) data base was developed to store and manipulate the large amount of information anticipated for the various Space Shuttle Main Engine (SSME) FIPMs. This report describes the organization and structure of the FIPM data base. This description includes a summary of the data fields and key attributes associated with each FIPM data file. The report also discusses the menu-driven software developed to facilitate and control the entry, modification, and listing of data base records. The final section of the report describes the transfer of the FIPM data base and software to the NASA Marshall Space Flight Center. The report appendixes include complete listings of all of the data base definition commands and software procedures.

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
FAILURE INFORMATION PROPAGATION MODEL	5
FIPM Definitions	5
FIPM Methodology	6
SSME FIPMs	8
FIPM DATA BASE	11
Data Base Structure	11
Data Description	16
Domains SYSTEMS and SYSTEMS_FORM	17
Domains MODULES and MODULES_FORM	20
Domains FAILUREMODES and FAILUREMODES_FORM	22
Domains CONNECTIONS and CONNECTIONS_FORM	24
Domains PROPAGATIONS_A150 through PROPAGATIONS_Z910 and PROPAGATIONS_FORM	26
Domains REFERENCES and REFERENCES_FORM	30
FIPM DATA BASE SOFTWARE	33
Digital Command Language Procedures	33
Datatrieve Command Files, Procedures, and Tables	41
Terminal Data Management System Forms	47
FIPM DATA BASE TRANSFER	49
REFERENCES	55
 APPENDIX A -- FIPM DOMAIN DEFINITIONS	 A-1
APPENDIX B -- FIPM RECORD DEFINITIONS	B-1
APPENDIX C -- FIPM FILE DEFINITIONS	C-1
APPENDIX D -- FIPM DCL COMMAND PROCEDURES	D-1
APPENDIX E -- FIPM DATATRIEVE COMMAND FILES	E-1
APPENDIX F -- FIPM DATATRIEVE PROCEDURES	F-1
APPENDIX G -- FIPM DATATRIEVE TABLES	G-1
APPENDIX H -- FIPM TDMS FORM DEFINITIONS	H-1

LIST OF TABLES

	<u>Page</u>
TABLE 1. FIPM RECORDS, DOMAINS, AND DATA FILES	15
TABLE 2. SUMMARY OF FIPM RECORD SYSTEMS_REC	18
TABLE 3. KEY FIELDS FOR DOMAINS SYSTEMS AND SYSTEMS_FORM	20
TABLE 4. SUMMARY OF FIPM RECORD MODULES_REC	21
TABLE 5. KEY FIELDS FOR DOMAINS MODULES AND MODULES_FORM	22
TABLE 6. SUMMARY OF FIPM RECORD FAILUREMODES_REC	23
TABLE 7. KEY FIELDS FOR DOMAINS FAILUREMODES AND FAILUREMODES_FORM	24
TABLE 8. SUMMARY OF FIPM RECORD CONNECTIONS_REC	25
TABLE 9. KEY FIELDS FOR DOMAINS CONNECTIONS AND CONNECTIONS_FORM	26
TABLE 10. SUMMARY OF FIPM RECORD PROPAGATIONS_REC	27
TABLE 11. KEY FIELDS FOR DOMAINS PROPAGATIONS_A150 THROUGH PROPAGATIONS_Z910 AND PROPAGATIONS_FORM	29
TABLE 12. SUMMARY OF FIPM RECORD REFERENCES_REC	31
TABLE 13. KEY FIELDS FOR DOMAINS REFERENCES AND REFERENCES_FORM . . .	32

LIST OF FIGURES

	<u>Page</u>
FIGURE 1. DATATRIEVE DEFINE DOMAIN COMMAND	12
FIGURE 2. DATATRIEVE DEFINE RECORD COMMAND	13
FIGURE 3. DATATRIEVE DEFINE FILE COMMAND	14
FIGURE 4. DOMAIN DEFINITION COMMANDS FOR PROPAGATIONS_A150	29
FIGURE 5. FILE DEFINITION COMMANDS FOR PROPAGATIONS_A150	30
FIGURE 6. MENU FOR CONTROLLED ACCESS TO FIPM DATA BASE	35
FIGURE 7. TOP-LEVEL FIPM SOFTWARE FLOWS	35
FIGURE 8. MENU FOR STORING FIPM DATA BASE RECORDS	37
FIGURE 9. PROGRAM FLOW FOR STORING FIPM DATA	37
FIGURE 10. MENU FOR MODIFYING FIPM DATA BASE RECORDS	38
FIGURE 11. PROGRAM FLOW FOR MODIFYING FIPM DATA	38
FIGURE 12. MENU FOR LISTING FIPM DATA BASE RECORDS	39
FIGURE 13. PROGRAM FLOW FOR LISTING FIPM DATA	40
FIGURE 14. SAMPLE DATATRIEVE TABLE	41
FIGURE 15. DATATRIEVE COMMAND FILES, PROCEDURES AND TABLES USED TO STORE FIPM DATA	43
FIGURE 16. DATATRIEVE COMMAND FILES, PROCEDURES AND TABLES USED TO MODIFY FIPM DATA	44
FIGURE 17. DATATRIEVE COMMAND FILES AND PROCEDURES USED TO LIST FIPM DATA	46
FIGURE 18. MISCELLANEOUS DATATRIEVE PROCEDURES AND TABLES USED FOR FIPM	46

LIST OF FIGURES
(Continued)

	<u>Page</u>
FIGURE 19. FIPM REQUEST LIBRARY DEFINITION	47
FIGURE 20. VAX/VMS FILES USED TO TRANSFER FIPM DATA BASE	49
FIGURE 21. FIPM DIRECTORY STRUCTURE	51
FIGURE 22. DIRECTORY DEV\$206:[BCDSSME2]	52
FIGURE 23. DIRECTORY DEV\$206:[BCDSSME2.DATA]	52
FIGURE 24. DIRECTORY DEV\$206:[BCDSSME2.DTR]	53
FIGURE 25. DIRECTORY DEV\$206:[BCDSSME2.FIPM]	53
FIGURE 26. DIRECTORY DEV\$206:[BCDSSME2.FORMS]	53

(This page intentionally blank)

STUDIES AND ANALYSES OF THE SPACE SHUTTLE MAIN ENGINE

TECHNICAL REPORT

on

FAILURE INFORMATION PROPAGATION MODEL DATA BASE AND SOFTWARE

Contract Number NASw-3737

INTRODUCTION

The failure information propagation model (FIPM) data base and software were part of an overall study of the Space Shuttle Main Engine (SSME) monitoring and diagnostic system. This study was conducted for the National Aeronautics and Space Administration, George C. Marshall Space Flight Center (NASA MSFC) under Contract No. NASw-3737. The principal tasks which comprised this study include:

- Review of the SSME failure data base to identify major failure types and to establish engine monitoring priorities
- Survey of diagnostic sensors, signal processing techniques, and monitoring systems associated with aerospace and other industries
- Systems-level analysis of the current SSME monitoring and diagnostic system using the outputs of the SSME failure data review and the diagnostic survey
- Recommendations concerning increased utilization of the current SSME monitoring/diagnostic data and potential improvements in the overall system.

The major emphasis of this study was to evaluate means for identifying and collecting high-quality data which maximizes knowledge of the overall engine condition. Information of this type is essential for both flight and ground test operations. The study also considers both real-time and post-operation processing of the collected data.

The SSME failure data review and the diagnostic survey were conducted in parallel during the initial phase of this study. These tasks provided valuable data on the engine, its operating characteristics, and

the general state of machine diagnostics. This information has been combined with the results of the current systems-level analysis of the SSME monitoring system to make recommendations concerning potential diagnostic improvements. The activities related to the SSME failure data review and the diagnostic survey are discussed in a previous technical report titled "Studies and Analyses of the Space Shuttle Main Engine, Technical Report Covering SSME Failure Data Review, Diagnostic Survey, and SSME Diagnostic Evaluation" (Reference 1). This document also includes a discussion of the early activities on the SSME diagnostic evaluation task.

The analysis tool selected for the evaluation of the engine monitoring system was the failure information propagation model (FIPM). The FIPM is a technique developed by Battelle to qualitatively analyze the information bearing value of all potential test points in a system. Initial attempts to create FIPMs for the high-pressure oxidizer turbopump (HPOTP) and the high-pressure fuel turbopump (HPFTP) demonstrated that large quantities of data would be associated with the various SSME models. This observation led to the decision to implement an interactive computer data base system to store and manipulate this information. The FIPM data base is one of two major elements in the current methodology. The data base serves as the repository for all of the data required to model and analyze given engine components. The FIPM drawing is the other principal element. The drawing is used primarily during generation of the information which is stored in the data base.

The FIPM data base was developed on a Digital Equipment Corporation (DEC) VAX computer. The software which controls the input, modification, and listing of data base records has been implemented using several different DEC software packages. This report documents the FIPM software developed during this study. The following items are included:

- Outline of the overall FIPM process
- General discussion of the FIPM data base
- Specific features of the FIPM data base software
- Listings of all the computer procedures, form definitions, etc.

The HPOTP was the first engine component to be modeled using this data base software. The HPOTP model was completed in February 1987. This FIPM is the subject of a separate technical report titled "Studies and Analyses of the Space Shuttle Main Engine, Technical Report on High-Pressure Oxidizer Turbopump Failure Information Propagation Model" (Reference 2). A copy of the FIPM software and the HPOTP data have been transferred to NASA MSFC.

(This page intentionally blank)

FAILURE INFORMATION PROPAGATION MODEL

The failure information propagation model (FIPM) is an analysis tool developed by Battelle's Columbus Division to systematically evaluate the potential test points in a system. The objective of this evaluation is to qualitatively assess the information bearing value of each test point. The FIPM methodology had demonstrated the capability to provide useful diagnostic insights for a broad range of mechanical and electronic systems in several previous studies. The FIPM was selected on this basis as the primary means for performing the SSME diagnostic assessment. It must be noted that the FIPM analyzes the propagation of failure information and not the actual failure. The model assumes that the system being depicted is in a near-normal state of operation. The failure information flow is described for the instant of time immediately following a given failure. Three principal applications exist for the output of this model. These applications are:

- Design of sensor systems for new devices or components
- Evaluation of existing sensor systems to maximize the information yield
- Identification of sensor research and development needed to target key diagnostic data.

This section briefly defines the terminology associated with an FIPM, describes the general features of the FIPM methodology, and discusses the specifics of applying the FIPM to analyze the SSME.

FIPM Definitions

The following terms are used in reference to a failure information propagation model:

- SYSTEM - The top-level item or component which is being modeled (analyzed)
- MODULE - A subelement or function of the system

- FAILURE MODE - The physical mechanism or process by which a module ceases to perform its intended function
- CONNECTION - A path (mechanical, fluid, etc.) which exists between two modules
- FAILURE INFORMATION PROPAGATION - A description of specific signal characteristics associated with a given failure mode which can be detected at a particular connection.

FIPM Methodology

The failure information propagation model basically divides the system under analysis into its constituent modules, describes the failure modes for each of the modules, catalogs the physical connections between the modules, details the flow of failure information through the various connections, and groups the failure information according to signal properties. An illustrative example of an exhaust fan FIPM is given in an earlier Battelle technical report (Reference 1). A series of guidelines, definitions, and rules have been developed to assist in the formulation of an FIPM. Two different approaches have been used to display and store FIPM related data. Each of these techniques is discussed briefly in this subsection.

The initial FIPM procedure was based entirely on a drawing or graphical representation of the system. All of the data associated with the model was shown on this drawing. This approach worked very well for the first three applications of the modeling technique (photographic copy machine, ion chamber, and home furnace). The limitations of this graphical FIPM approach were demonstrated during subsequent attempts to model more complex mechanical systems such as the SSME. The major problem is the excessive amount of data which must be displayed while maintaining reasonable constraints on physical size. It is also very difficult to adequately differentiate all of the various failure signals and characteristics within the context of a graphical representation.

The current FIPM methodology consists of two primary elements. These elements are:

- Simplified FIPM drawing
- FIPM data base.

The present FIPM drawing format summarizes key information about the system being modeled for use during generation and input of appropriate data base records. The data base stores all of the information associated with the FIPM. This includes the items shown on the drawing plus a substantial amount of supplemental information and comments. The data base is the only location where the actual failure information propagations are stored. The FIPM data base is described in a subsequent section of this report. The FIPM drawing is briefly discussed in the following paragraphs.

The first step in formulating a failure information propagation model is to develop a graphical representation or drawing of the system being analyzed. The principal function of the FIPM drawing is to describe the constituent modules of the system and to identify the connections between these modules. The initial drafts of the FIPM drawing are prepared by technical analysts or engineers familiar with the system involved. The number of modules included is chosen to be consistent with the overall level of detail required for the analysis. The accurate depiction of the system is critical to the overall development of the FIPM. This illustration is the foundation for the entire data base associated with a given system. Careful construction and review of the FIPM drawing minimizes potential corrections and changes to the data base.

The FIPM drawing is composed basically of boxes and lines which connect the boxes. Each box on the drawing represents a particular module. The lines represent the physical connections between the various modules. Additional information is also shown for both the boxes (modules) and the lines (connections) to further identify specific physical details associated with both of these elements. Each module on the FIPM drawing displays the following items of information: system code, module number, module name, and module failure modes. Examination of the line type and symbols associated with specific connections enables the following items of information to be determined: general type of connection (solid, liquid, etc.), additional data on the exact type of connection (common-piece, oxygen, etc.), unanticipated connection, and

connection to external system. Symbols are combined as required to completely describe a particular connection. The format selected for the FIPM drawing allows all of the necessary data to be displayed in black and white for ease of reproduction.

SSME FIPMs

The Space Shuttle Main Engine is the most complex mechanical system evaluated to date using the failure information propagation model. The initial approach to analyzing the SSME divides the engine into major components (systems) which are examined independently. This process reduces the size of the individual models to a manageable level and also eliminates the crossflow of failure information between systems. The FIPM for each of the engine elements can be analyzed to yield monitoring recommendations for that particular component. After studying each of the major systems, the individual results can be integrated to yield a set of diagnostic requirements and recommendations for the entire SSME.

The "SSME Failure Mode and Effects Analysis (FMEA) and Critical Items List" compiled by the Rocketdyne Division, Rockwell International Corporation (Reference 3) includes over 200 SSME components. Developing an individual FIPM for each of these items would not be the most efficient way to analyze the entire engine. Certain components, such as propellant ducts and pressurant lines, are relatively simple in nature. These systems can be easily modeled with just a few modules and connections. SSME items of this type are included as modules in the FIPM of the appropriate major component. For example, the high-pressure oxidizer duct is included with the HPOTP FIPM.

Each system (major component) is represented in the FIPM data base by a four-digit code. These system designations coincide with the Rocketdyne FMEA item numbers (Reference 3, Table 2-1) whenever feasible. The record which defines the system also indicates any additional Rocketdyne FMEA items which are included in that particular FIPM. Components or systems which do not have a corresponding Rocketdyne FMEA number are given a similar four-digit code. Confusion is avoided by selecting a number not used by Rocketdyne.

The FIPM methodology, as used for analyzing the SSME, includes special provisions for handling the connections between major engine components (FIPM systems). This feature of the technique allows the data flows between systems to be evaluated on a preliminary basis. It also enables the future expansion of the SSME model to a higher level through the combination of various system FIPMs.

(This page intentionally blank)

FIPM DATA BASE

The FIPM data base is a computerized system which stores all of the data necessary to create the various SSME failure information propagation models. The information contained in the FIPM data base is divided into the following six categories: systems, modules, connections, failure modes, failure information propagations, and references. Each of these categories corresponds to a major element of the overall FIPM process as discussed in the previous section. The data base was designed to store the essential FIPM information, additional descriptive data pertinent to each category and entries which document data base operations. Details on the structure and contents of the FIPM data base are included in the following subsections.

The FIPM data base has been implemented on a Digital Equipment Corporation (DEC) VAX computer. The data base management system selected was DEC's VAX Datatrieve. The computer and data base system were selected based on the availability of these items at both Battelle and NASA MSFC. The data base design and development activities were performed on the Battelle computers. After entry and verification at Battelle, the initial FIPM data files were transferred to NASA MSFC in February 1987.

Data Base Structure

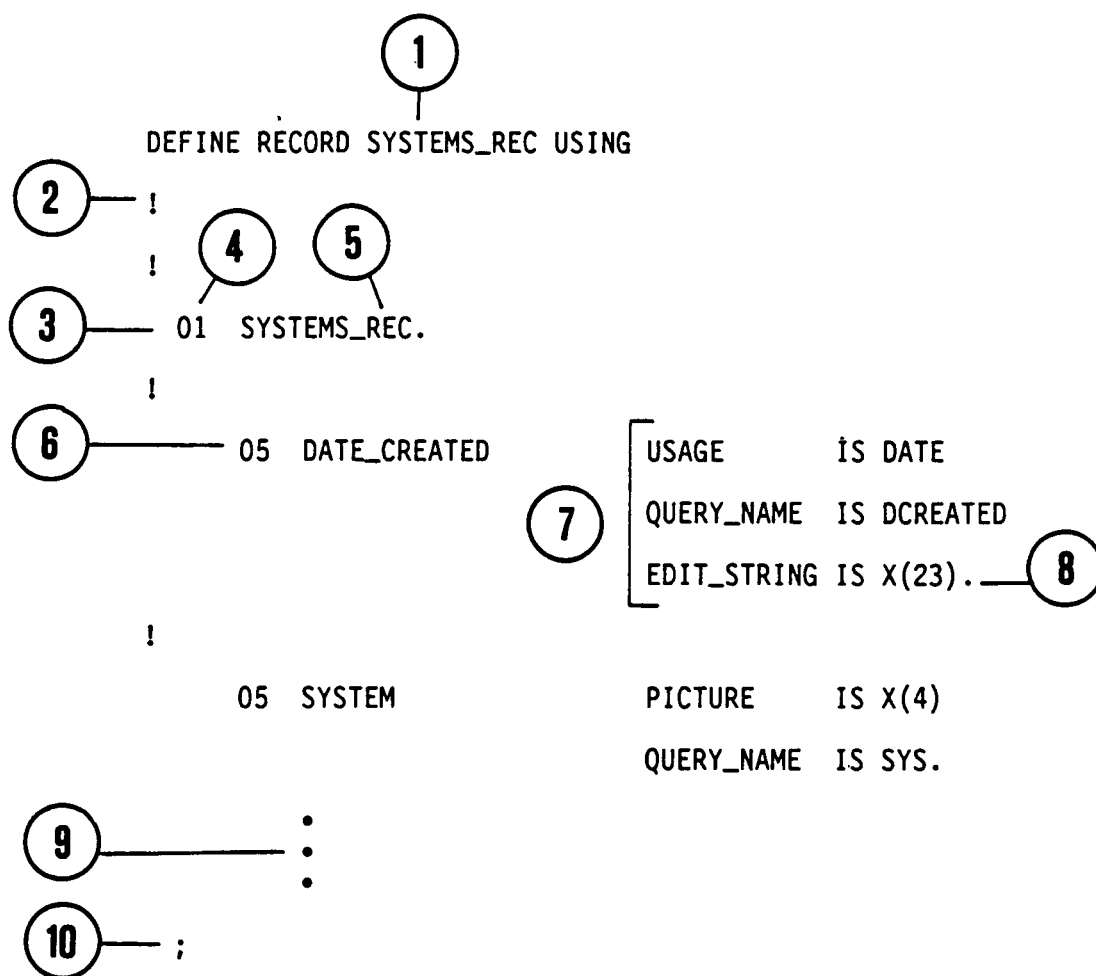
The fundamental elements required to create a Datatrieve data base are records, domains, and data files. Records are the detailed descriptions of the data fields (information) which are stored. Domains are sets of data which share a common record definition. The data files are the actual VAX RMS (record management services) files which contain the information. Each of these elements must be defined at the Datatrieve command level before information can be stored. A domain is logically related to the corresponding record and data file through the domain definition. The definition for one of the FIPM domains (SYSTEMS) is shown in Figure 1. An excerpt from the corresponding record definition (SYSTEMS_REC) is displayed in Figure 2. The file definition command for this domain is illustrated in Figure 3.

DEFINE DOMAIN SYSTEMS USING SYSTEMS_REC ON
DEV\$206:[BCDSSME2.DATA]SYSTEMS.DAT
; —

The diagram consists of four numbered circles with lines pointing to specific parts of the command text. Circle 1 points to 'SYSTEMS' in the first line. Circle 2 points to 'SYSTEMS_REC' in the first line. Circle 3 points to 'SYSTEMS.DAT' in the second line. Circle 4 points to the semicolon ';' in the third line.

- 1 - Domain name
- 2 - Record name
- 3 - Data file
- 4 - Definition terminator

FIGURE 1. DATATRIEVE DEFINE DOMAIN COMMAND



- 1 - Record name
- 2 - Comment line
- 3 - Group field
- 4 - Level number
- 5 - Field name
- 6 - Elementary field
- 7 - Field definition clauses
- 8 - Field definition terminator
- 9 - Additional field definitions
- 10 - Record definition terminator

FIGURE 2. DATATRIEVE DEFINE RECORD COMMAND

①
 DEFINE FILE FOR SYSTEMS KEY = DATE_CREATED
 KEY = SYSTEM
 KEY = SYSTEM_NAME
 (DUP), — ③
] ④

The diagram shows the command 'DEFINE FILE FOR SYSTEMS' with a line from circle 1 pointing to it. The key clauses are 'KEY = DATE_CREATED', 'KEY = SYSTEM', and 'KEY = SYSTEM_NAME'. The first key clause is followed by '(DUP), —' with a line from circle 2 pointing to '(DUP)' and a line from circle 3 pointing to the comma. The last key clause is followed by a closing bracket ']' with a line from circle 4 pointing to it.

- 1 - Domain name
- 2 - Field option to allow duplicate values
- 3 - Primary key clause
- 4 - Secondary key clauses

FIGURE 3. DATATRIEVE DEFINE FILE COMMAND

The FIPM data base is structured around six Datatrieve records. These include:

- SYSTEMS_REC
- MODULES_REC
- FAILUREMODES_REC
- CONNECTIONS_REC
- PROPAGATIONS_REC
- REFERENCES_REC.

Each of the records SYSTEMS_REC, MODULES_REC, FAILUREMODES_REC, CONNECTIONS_REC, and REFERENCES_REC is associated with two FIPM domains. PROPAGATIONS_REC is the basis for a group of domains which store SSME failure information propagation records. Table 1 lists all of the FIPM records, domains, and data files.

TABLE 1. FIPM RECORDS, DOMAINS, AND DATA FILES

Record	Domain	Data File
SYSTEMS_REC	SYSTEMS SYSTEMS_FORM	DEV\$206:[BCDSSME2.DATA]SYSTEMS.DAT DEV\$206:[BCDSSME2.DATA]SYSTEMS_FORM.DAT
MODULES_REC	MODULES MODULES_FORM	DEV\$206:[BCDSSME2.DATA]MODULES.DAT DEV\$206:[BCDSSME2.DATA]MODULES_FORM.DAT
FAILUREMODES_REC	FAILUREMODES FAILUREMODES_FORM	DEV\$206:[BCDSSME2.DATA]FAILUREMODES.DAT DEV\$206:[BCDSSME2.DATA]FAILUREMODES_FORM.DAT
CONNECTIONS_REC	CONNECTIONS CONNECTIONS_FORM	DEV\$206:[BCDSSME2.DATA]CONNECTIONS.DAT DEV\$206:[BCDSSME2.DATA]CONNECTIONS_FORM.DAT
PROPAGATIONS_REC	PROPAGATIONS_A150 . . .	DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_A150.DAT . . .
	PROPAGATIONS_Z910 PROPAGATIONS_FORM	DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_Z910.DAT DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_FORM.DAT
REFERENCES_REC	REFERENCES REFERENCES_FORM	DEV\$206:[BCDSSME2.DATA]REFERENCES.DAT DEV\$206:[BCDSSME2.DATA]REFERENCES_FORM.DAT

The failure information propagations are not stored in a single domain (data file) due to the large number of data records involved. In the case of the HPOTP FIPM, there are 8213 failure information propagations. The access time for large files is a limiting factor on the overall size of the file. Experience with the HPOTP model indicated that a separate failure information propagation domain should be created for each SSME system (major component) being modeled. This format was adopted for the FIPM data base. As mentioned earlier, the same Datatrieve record definition is used for all of the propagations domains.

The data file associated with each FIPM domain is a VAX RMS indexed sequential file. These files contain an index of pointers based on the specified primary and secondary keys. The index allows the file access system to rapidly locate a record with specific attributes. This feature significantly improves the time required for many input and output operations. A primary key and at least one secondary key have been defined for all of the FIPM domains.

Data Description

The data formats established for the various FIPM domains are described in the following subsections. The data fields, query names, field type, data class, field length, and total record length are discussed for each of the Datatrieve domains. A query name is an abbreviated form of the field name which can be used during Datatrieve operations. The field type is group, elementary, or redefines. A group field contains one or more additional fields while an elementary field contains a single item of data. A redefines field creates an alternate definition for either a group or an elementary field without increasing the total length of the record. The field class describes the nature of the data contained in that field. Three field classes were used to define the six FIPM records: alphanumeric, numeric, and date. An alphanumeric field can contain any member of the Datatrieve character set (letter, digit, or special). A numeric field is restricted to digits plus an optional sign (+ or -). The date field is required for storing and manipulating dates in Datatrieve. The key fields which have been defined for the respective VAX RMS files are also identified.

Domains SYSTEMS and SYSTEMS_FORM

The domain SYSTEMS stores information which defines the top-level items or components (systems) being modeled. Each major engine component (high-pressure oxidizer turbopump, oxidizer preburner, etc.) has a corresponding FIPM system. A field has been provided for storing a descriptive name for each system. A total of 15 fields have been defined for identifying the Rocketdyne FMEA items which comprise each system. Fields also are included for specifying reference documents which were used in formulating each system model. Several additional fields are defined for storage of pertinent data relative to the creation and modification of each record. Domain SYSTEMS_FORM is used to display input/output forms on the computer terminal during data entry and modification. It is functionally identical to SYSTEMS but contains only one record.

The domain definitions for SYSTEMS and SYSTEMS_FORM are included in Appendix A. The corresponding record definition, SYSTEMS_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 2. The Datatrieve file definition commands for both of these domains are included in Appendix C. The key fields for SYSTEMS and SYSTEMS_FORM are given in Table 3.

TABLE 2. SUMMARY OF FIPM RECORD SYSTEMS_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
SYSTEMS_REC	--	Group	Alphanumeric	--	250
DATE_CREATED	DCREATED	Elementary	Date	--	8
SYSTEM	SYS	Elementary	Alphanumeric	4	4
SYSTEM_NAME	SYSNAME	Elementary	Alphanumeric	80	80
FMEA_ITEMS	--	Group	Alphanumeric	--	60
ITEM1	--	Elementary	Alphanumeric	4	4
ITEM2	--	Elementary	Alphanumeric	4	4
ITEM3	--	Elementary	Alphanumeric	4	4
ITEM4	--	Elementary	Alphanumeric	4	4
ITEM5	--	Elementary	Alphanumeric	4	4
ITEM6	--	Elementary	Alphanumeric	4	4
ITEM7	--	Elementary	Alphanumeric	4	4
ITEM8	--	Elementary	Alphanumeric	4	4
ITEM9	--	Elementary	Alphanumeric	4	4
ITEM10	--	Elementary	Alphanumeric	4	4
ITEM11	--	Elementary	Alphanumeric	4	4
ITEM12	--	Elementary	Alphanumeric	4	4
ITEM13	--	Elementary	Alphanumeric	4	4
ITEM14	--	Elementary	Alphanumeric	4	4
ITEM15	--	Elementary	Alphanumeric	4	4

TABLE 2. SUMMARY OF FIPM RECORD SYSTEMS_REC (CONTINUED)

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
REFERENCES	--	Group	Alphanumeric	--	50
REFERENCE1	REF1	Elementary	Alphanumeric	5	5
REFERENCE2	REF2	Elementary	Alphanumeric	5	5
REFERENCE3	REF3	Elementary	Alphanumeric	5	5
REFERENCE4	REF4	Elementary	Alphanumeric	5	5
REFERENCE5	REF5	Elementary	Alphanumeric	5	5
REFERENCE6	REF6	Elementary	Alphanumeric	5	5
REFERENCE7	REF7	Elementary	Alphanumeric	5	5
REFERENCE8	REF8	Elementary	Alphanumeric	5	5
REFERENCE9	REF9	Elementary	Alphanumeric	5	5
REFERENCE10	REF10	Elementary	Alphanumeric	5	5
PROPAGATIONS_FILE_CREATED	FIPCREATED	Elementary	Alphanumeric	3	3
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	17	17

TABLE 3. KEY FIELDS FOR DOMAINS SYSTEMS AND SYSTEMS_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
SYSTEM	Alternate	Yes	Yes
SYSTEM_NAME	Alternate	Yes	Yes

Domains MODULES and MODULES_FORM

The domain MODULES stores information which defines the subelements or functions comprising each of the systems (SSME components) being modeled. Each FIPM system has multiple modules which are identified by the combination of the system and a unique module number. Fields have been included for storing a descriptive name and the general function associated with each module. Several additional fields also are defined for storage of pertinent data relative to the creation and modification of each record. Domain MODULES_FORM is used to display input/output forms on the computer terminal during data entry and modification. It is functionally identical to MODULES but contains only one record.

The domain definitions for MODULES and MODULES_FORM are included in Appendix A. The corresponding record definition, MODULES_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 4. The Datatrieve file definition commands for both of these domains are included in Appendix C. The key fields for MODULES and MODULES_FORM are given in Table 5.

TABLE 4. SUMMARY OF FIPM RECORD MODULES_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
MODULES_REC	--	Group	Alphanumeric	--	406
DATE_CREATED	DCREATED	Elementary	Date	--	8
SYSTEM_MODULE	SYSMOD	Elementary	Alphanumeric	8	8
SYSTEM_MODULE_PARTS	--	Redefines	--	--	--
SYSTEM_MODULE	SYS	Elementary	Alphanumeric	4	4
MODULE	MOD	Elementary	Numeric	4	4
SYSTEM_MODULE_NAME	SYSMODNAME	Elementary	Alphanumeric	80	80
SYSTEM_MODULE_FUNCTION	SYSMODFUNC	Elementary	Alphanumeric	242	242
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	40	40

TABLE 5. KEY FIELDS FOR DOMAINS MODULES AND MODULES_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
SYSTEM_MODULE	Alternate	Yes	Yes
SYSTEM_MODULE_NAME	Alternate	Yes	Yes

Domains FAILUREMODES and FAILUREMODES_FORM

The domain FAILUREMODES stores information which defines the failure modes identified for each module. The individual modules, in general, will have multiple failure modes. The principal field for each record is a 20-character code which specifies the source module, the failure mechanism, and any accomplice module which may be involved. Fields are provided for the entry of text which describes the failure mode and identifies the general effects associated with it. Several additional fields also are defined for storage of pertinent data relative to the creation and modification of each record. Domain FAILUREMODES_FORM is used to display input/output forms on the computer terminal during data entry and modification. It is functionally identical to FAILUREMODES but contains only one record.

The domain definitions for FAILUREMODES and FAILUREMODES_FORM are included in Appendix A. The corresponding record definition, FAILUREMODES_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 6. The Datatrieve file definition commands for both of these domains are included in Appendix C. The key fields for FAILUREMODES and FAILUREMODES_FORM are given in Table 7.

TABLE 6. SUMMARY OF FIPM RECORD FAILUREMODES_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
FAILUREMODES_REC	--	Group	Alphanumeric	--	1364
DATE_CREATED	DCREATED	Elementary	Date	--	8
FMCODE	--	Elementary	Alphanumeric	20	20
FMCODE_PARTS	--	Redefines	--	--	--
SOURCE_SYSTEM_MODULE	SSYSMOD	Elementary	Alphanumeric	8	8
SOURCE_SYSTEM_MODULE_PARTS	--	Redefines	--	--	--
SOURCE_SYSTEM	SSYS	Elementary	Alphanumeric	4	4
SOURCE_MODULE	SMOD	Elementary	Numeric	4	4
FAILURE_MODE_SUBMODE	FMSUBM	Elementary	Alphanumeric	4	4
FAILURE_MODE_SUBMODE_PARTS	--	Redefines	--	--	--
FAILURE_MODE	FM	Elementary	Alphanumeric	2	2
FAILURE_SUBMODE	FSUBM	Elementary	Alphanumeric	2	2
ACCOMPLICE_SYSTEM_MODULE	ACCSYSMOD	Elementary	Alphanumeric	8	8
ACCOMPLICE_SYSTEM_MODULE_PARTS	--	Redefines	--	--	--
ACCOMPLICE_SYSTEM	ACCSYS	Elementary	Alphanumeric	4	4
ACCOMPLICE_MODULE	ACCMOD	Elementary	Numeric	4	4
DESCRIPTION	DESC	Elementary	Alphanumeric	242	242
EFFECTS	--	Group	Alphanumeric	--	966
EFFECT1	--	Elementary	Alphanumeric	161	161
EFFECT2	--	Elementary	Alphanumeric	161	161
EFFECT3	--	Elementary	Alphanumeric	161	161
EFFECT4	--	Elementary	Alphanumeric	161	161
EFFECT5	--	Elementary	Alphanumeric	161	161
EFFECT6	--	Elementary	Alphanumeric	161	161
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	30	100

TABLE 7. KEY FIELDS FOR DOMAINS FAILUREMODES
AND FAILUREMODES_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
FMCODE	Alternate	Yes	Yes

Domains CONNECTIONS and CONNECTIONS_FORM

The domain CONNECTIONS stores information which defines the physical paths which exist between modules. In general, each module will have multiple connections to the adjacent module(s). The principal field in each record is a 21-character code which specifies the two modules being connected and the exact nature of the connection. Several additional fields also are defined for storage of pertinent data relative to the creation and modification of each record. Domain CONNECTIONS_FORM is used to display input/output forms on the computer terminal during data entry and modification. It is functionally identical to CONNECTIONS but contains only one record.

The domain definitions for CONNECTIONS and CONNECTIONS_FORM are included in Appendix A. The corresponding record definition, CONNECTIONS_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 8. The Datatrieve file definition commands for both of these domains are included in Appendix C. The key fields for CONNECTIONS and CONNECTIONS_FORM are given in Table 9.

TABLE 8. SUMMARY OF FIPM RECORD CONNECTIONS_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
CONNECTIONS_REC	--	Group	Alphanumeric	--	77
DATE_CREATED	DCREATED	Elementary	Date	--	8
CODE_NUMBER	CODENO	Elementary	Alphanumeric	21	21
CODE_NUMBER_PARTS	--	Redefines	--	--	--
SYSTEM_MODULE_A	SYSMODA	Elementary	Alphanumeric	8	8
SYSTEM_MODULE_A_PARTS	--	Redefines	--	--	--
SYSTEM_A	SYSA	Elementary	Alphanumeric	4	4
MODULE_A	MODA	Elementary	Numeric	4	4
CONNECTION	CN	Elementary	Alphanumeric	4	4
CONNECTION_PARTS	--	Redefines	--	--	--
CONNECTION_TYPE	CNTYPE	Elementary	Alphanumeric	2	2
CONNECTION_QUALIFIER	CNQUAL	Elementary	Alphanumeric	2	2
UNANTICIPATED_CONNECTION	UA	Elementary	Alphanumeric	1	1
SYSTEM_MODULE_B	SYSMODB	Elementary	Alphanumeric	8	8
SYSTEM_MODULE_B_PARTS	--	Redefines	--	--	--
SYSTEM_B	SYSB	Elementary	Alphanumeric	4	4
MODULE_B	MODB	Elementary	Numeric	4	4
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	30	20

TABLE 9. KEY FIELDS FOR DOMAINS CONNECTIONS
AND CONNECTIONS_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
CODE_NUMBER	Alternate	Yes	Yes

Domains PROPAGATIONS_A150 through PROPAGATIONS_Z910
and PROPAGATIONS_FORM

The domains PROPAGATIONS_A150 through PROPAGATIONS_Z910 store the actual failure information propagation data. Each of the items in domain SYSTEMS has a separate propagations domain. One of the fields identifies the module failure mode which initiated the information flow. Another field specifies the particular connection to which the data has passed. Most of the fields describe the specific characteristics of the failure signal. Three text fields have been included for entry of comments pertaining to the failure information propagation. Three fields also are defined for storage of data concerning the creation and modification of each record. Domain PROPAGATIONS_FORM is used to display input/output forms on the computer terminal. It is functionally identical to the other propagations domains but contains only one record.

The domain definitions for all of the current failure information propagations domains are included in Appendix A. The corresponding record definition, PROPAGATIONS_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 10. The Datatrieve file definition commands for all of the domains are included in Appendix C. The key fields for PROPAGATIONS_A150 through PROPAGATIONS_Z910 and PROPAGATIONS_FORM are given in Table 11.

TABLE 10. SUMMARY OF FIPM RECORD PROPAGATIONS_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
PROPAGATIONS_REC					473
DATE_CREATED	-- DCREATED	Elementary	Alphanumeric	--	8
FMCODE	--	Elementary	Alphanumeric	20	20
FMCODE_PARTS	--	Elementary	Alphanumeric	--	--
SOURCE_SYSTEM_MODULE	-- SSYSMOD	Elementary	Alphanumeric	8	8
SOURCE_SYSTEM_MODULE_PARTS	--	Elementary	Alphanumeric	--	--
SOURCE_SYSTEM_MODULE	-- SSYS	Elementary	Alphanumeric	4	4
SOURCE_MODULE	-- SMOD	Elementary	Numeric	4	4
FAILURE_MODE_SUBMODE	-- FMSUBM	Elementary	Alphanumeric	4	4
FAILURE_MODE_SUBMODE_PARTS	--	Elementary	Alphanumeric	--	--
FAILURE_MODE	-- FM	Elementary	Alphanumeric	2	2
FAILURE_SUBMODE	-- FSUBM	Elementary	Alphanumeric	2	2
ACCOMPLICE_SYSTEM_MODULE	-- ACCSYSMOD	Elementary	Alphanumeric	8	8
ACCOMPLICE_SYSTEM_MODULE_PARTS	--	Elementary	Alphanumeric	--	--
ACCOMPLICE_SYSTEM_MODULE	-- ACCSYS	Elementary	Alphanumeric	4	4
ACCOMPLICE_SYSTEM_MODULE	-- ACCMOD	Elementary	Numeric	4	4
CODE_NUMBER	-- CODENO	Elementary	Alphanumeric	21	21
CODE_NUMBER_PARTS	--	Elementary	Alphanumeric	--	--
SYSTEM_MODULE_A	-- SYSMODA	Elementary	Alphanumeric	8	8
SYSTEM_MODULE_A_PARTS	--	Elementary	Alphanumeric	--	--
SYSTEM_A	-- SYSA	Elementary	Alphanumeric	4	4
MODULE_A	-- MODA	Elementary	Numeric	4	4
CONNECTION	-- CN	Elementary	Alphanumeric	4	4
CONNECTION_PARTS	--	Elementary	Alphanumeric	--	--
CONNECTION_TYPE	-- CNTYPE	Elementary	Alphanumeric	2	2
CONNECTION_QUALIFIER	-- CNQUAL	Elementary	Alphanumeric	2	2
UNANTICIPATED_CONNECTION	-- UA	Elementary	Alphanumeric	1	1
SYSTEM_MODULE_B	-- SYSMODB	Elementary	Alphanumeric	8	8
SYSTEM_MODULE_B_PARTS	--	Elementary	Alphanumeric	--	--
SYSTEM_B	-- SYSB	Elementary	Alphanumeric	4	4
MODULE_B	-- MODB	Elementary	Numeric	4	4

TABLE 10. SUMMARY OF FIPM RECORD PROPAGATIONS_REC (CONTINUED)

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
SIGNAL_DESCRIPTION	--	Group	Alphanumeric	--	366
RAW_SIGNAL	--	Group	Alphanumeric	--	76
SIGNAL_TYPE	SIG	Elementary	Alphanumeric	20	20
SIGNAL_UNITS	SIGUNIT	Elementary	Alphanumeric	25	25
DIMENSIONS	DIM	Elementary	Numeric	1	1
SIGNAL_QUALITY	SIGQUAL	Elementary	Numeric	1	1
FREQUENCY_TIME	--	Group	Alphanumeric	--	29
MAX_FREQ_OR_TIME	MAXFT	Elementary	Numeric	2	2
MIN_FREQ_OR_TIME	MINFT	Elementary	Numeric	2	2
FT_UNITS	FTUNIT	Elementary	Alphanumeric	25	25
SYMPOM_ELEMENT	--	Group	Alphanumeric	--	50
SENSITIVE_PARAMETER	--	Group	Alphanumeric	--	45
PARAMETER	PAR	Elementary	Alphanumeric	20	20
PARAMETER_UNITS	PARUNIT	Elementary	Alphanumeric	25	25
SYMPTOM_DURATION	SYMDUR	Elementary	Numeric	2	2
PERIOD_OF_ONSET	ONSET	Elementary	Numeric	2	2
INDICATES_FAILURE	INDFAIL	Elementary	Alphanumeric	1	1
COMMENTS	--	Group	Alphanumeric	--	240
COMMENT1	--	Elementary	Alphanumeric	80	80
COMMENT2	--	Elementary	Alphanumeric	80	80
COMMENT3	--	Elementary	Alphanumeric	80	80
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	30	30

TABLE 11. KEY FIELDS FOR DOMAINS PROPAGATIONS_A150
THROUGH PROPAGATIONS_Z910 AND
PROPAGATIONS_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
FMCODE	Alternate	Yes	Yes
CODE_NUMBER	Alternate	Yes	Yes
SIGNAL_TYPE	Alternate	Yes	Yes

The domain and file definition commands for PROPAGATIONS_A150 through PROPAGATIONS_Z910 differ from those used for the other FIPM domains. The domain and file definition commands associated with PROPAGATIONS_A150 are shown respectively in Figures 4 and 5. The domain definition uses the Datatrieve logicals PROPAGATIONS and PROPAGATIONS_FILE while the file definition uses the logical PROPAGATIONS. This process was selected to allow automated definition of a failure information propagation domain and file for each new entry in domain SYSTEMS.

```
FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A150")
FN$CREATE_LOG("PROPAGATIONS_FILE",
  "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_A150.DAT")
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE
;
```

FIGURE 4. DOMAIN DEFINITION COMMANDS FOR PROPAGATIONS_A150


```
FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A150")
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),
                              KEY = FMCODE        (DUP),
                              KEY = CODE_NUMBER   (DUP),
                              KEY = SIGNAL_TYPE   (DUP)
```

FIGURE 5. FILE DEFINITION COMMANDS FOR PROPAGATIONS_A150

Domains REFERENCES and REFERENCES_FORM

The domain REFERENCES stores information on the various documents used during the formulation of the FIPMs. The fields in this record provide for the input of standard bibliographical information such as author(s), title, company, company document number, data, and contract number. Another field stores a unique reference number for the document which is assigned by the Datatrieve input procedure. Several additional fields also are defined for storage of pertinent data relative to the creation and modification of each record. Domain REFERENCES_FORM is used to display input/output forms on the computer terminal during data entry and modification. It is functionally identical to REFERENCES but contains only one record.

The domain definitions for REFERENCES and REFERENCES_FORM are included in Appendix A. The corresponding record definition, REFERENCES_REC, is contained in Appendix B. The major features associated with this record are summarized in Table 12. The Datatrieve file definition commands for both of these domains are included in Appendix C. The key fields for REFERENCES and REFERENCES_FORM are given in Table 13.

TABLE 12. SUMMARY OF FIPM RECORD REFERENCES_REC

Field	Query Name	Type	Class	Number of Digits or Characters	Length (bytes)
REFERENCES_REC	--	Group	Alphanumeric	--	433
DATE_CREATED	DCREATED	Elementary	Date	--	8
REFERENCE_NUMBER	REFNO	Elementary	Alphanumeric	5	5
REFERENCE_NUMBER_PARTS	--	Redefines	--	--	--
SOURCE_ABBREVIATION	SABBREV	Elementary	Alphanumeric	2	2
SEQUENCE_NUMBER	SEQNO	Elementary	Numeric	3	3
AUTHORS	--	Group	Alphanumeric	--	100
AUTHOR1	--	Elementary	Alphanumeric	25	25
AUTHOR2	--	Elementary	Alphanumeric	25	25
AUTHOR3	--	Elementary	Alphanumeric	25	25
AUTHOR4	--	Elementary	Alphanumeric	25	25
DOCUMENT_TITLE	TITLE	Elementary	Alphanumeric	161	161
DOCUMENT_SOURCE	SOURCE	Elementary	Alphanumeric	30	30
DOCUMENT_NUMBER	DOCNO	Elementary	Alphanumeric	30	30
DOCUMENT_DATE	DOCDATE	Elementary	Alphanumeric	11	11
CONTRACT_NUMBER	CONTNO	Elementary	Alphanumeric	20	20
DATE_LAST_MODIFIED	DLASTMOD	Elementary	Date	--	8
MODIFYING_PROCEDURE	MODPROC	Elementary	Alphanumeric	20	20
FILLER	--	Elementary	Alphanumeric	40	40

TABLE 13. KEY FIELDS FOR DOMAINS REFERENCES AND REFERENCES_FORM

Field	Key Type	Duplicate Values	Change Values
DATE_CREATED	Primary	Yes	No
REFERENCE_NUMBER	Alternate	Yes	Yes
DOCUMENT_TITLE	Alternate	Yes	Yes
DOCUMENT_SOURCE	Alternate	Yes	Yes

FIPM DATA BASE SOFTWARE

The FIPM data base development software provides a controlled, interactive environment in which failure information propagation data can be stored, modified, and listed. The software allows the user to perform a number of predefined data base operations. Direct access to the data base is restricted to prevent inadvertent changes which can invalidate large portions of the data files. The software also performs an extensive number of validation tests on the information entered by the user during the storage and modification of FIPM records. The data base software was developed using the following three Digital Equipment Corporation (DEC) software packages:

- VAX/VMS Digital Command Language (DCL)
- Datatrieve
- Terminal Data Management System (TDMS).

DCL command procedures provide the overall control of the FIPM software through a series of four menus. VAX command files containing Datatrieve instructions are used in conjunction with the menus to initiate the storage, modification, or listing of FIPM information. The actual manipulation of the FIPM records is accomplished using Datatrieve procedures and tables. Terminal forms created using TDMS utilities provide the interactive user interface. The DCL, Datatrieve, and TDMS software elements are outlined in the following subsections.

Digital Command Language Procedures

The Digital Command Language enables the user to instruct the VAX/VMS operating system to perform various operations. DCL command procedures are files which contain a series of DCL commands. When a command procedure is executed, the computer processes all of the commands contained in the file and then returns to the point of origin. DCL command procedures are used in the FIPM data base to provide the top-level control of the software elements.

When a user initiates a VAX computer session, the operating system searches the default file directory for a file named LOGIN.COM. If the file is found, the computer executes the DCL commands in LOGIN.COM before performing any other operations. The FIPM data base development software uses this intrinsic VAX process to direct the program flow into a carefully controlled environment. The user is channeled from one procedure to the next without going to the DCL command level. Provisions are incorporated for users with special access privileges to bypass these procedures and execute commands at the DCL level.

The LOGIN.COM file created for the FIPM data base pauses for a response from the terminal. If the user enters the correct access code, the procedure will prompt for PASSWORD 1 and then PASSWORD 2. The procedure exits to the DCL command level if the access code and both passwords are entered correctly. If either PASSWORD 1 or PASSWORD 2 is not valid, the procedure loops back to the point of the initial pause. All responses except for the correct access code will result in the computer executing the DCL procedure FIPM_MENU.COM. FIPM_MENU.COM displays the main FIPM menu to the user. This menu is shown in Figure 6. The program flow is directed to either FIPM_STORE.COM, FIPM_MODIFY.COM, or FIPM_LIST.COM depending on the line number selected (1, 2, or 3 respectively). The user can also terminate the current computer session by entering line number 4. It is possible to exit to the DCL command level from the main menu by entering the correct access code and passwords. The top-level FIPM software flow is depicted in Figure 7. Listings of the DCL procedures LOGIN.COM and FIPM_MENU.COM are included in Appendix D.

=====

FAILURE INFORMATION PROPAGATION MODEL

=====

MAIN MENU

1. Store FIPM Data
2. Modify FIPM Data
3. List FIPM Data
4. Exit Procedure and Logout

Please enter LINE NUMBER:

FIGURE 6. MENU FOR CONTROLLED ACCESS TO FIPM DATA BASE

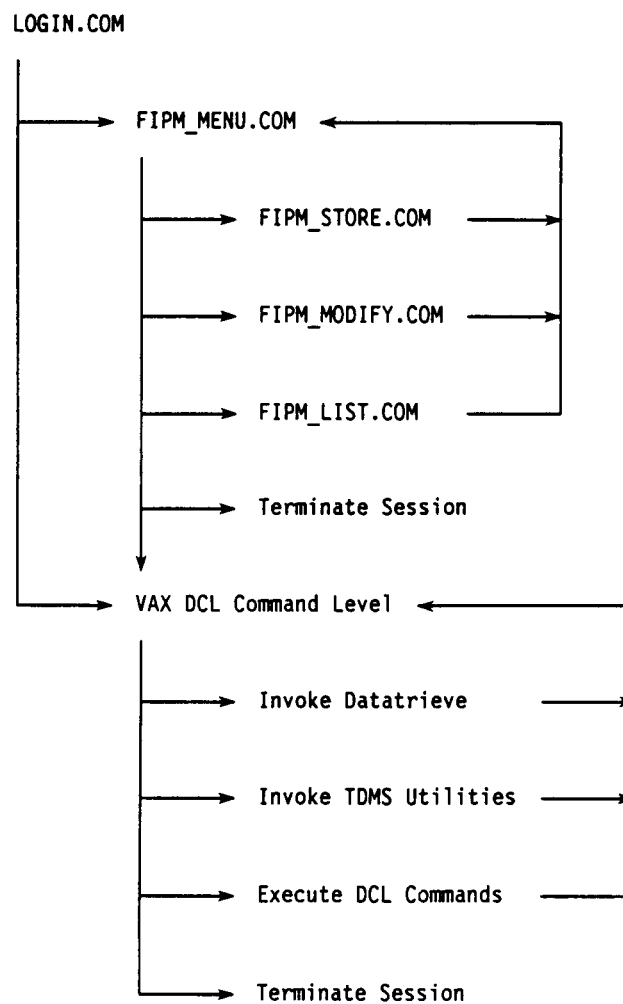


FIGURE 7. TOP-LEVEL FIPM SOFTWARE FLOWS

If the user selects the store FIPM data option, the DCL procedure FIPM_STORE.COM is called to display the menu shown in Figure 8. A response of 1 through 6 will result in the execution of the Datatrieve command file STORE_REF.COM, STORE_SYS.COM, STORE_MOD.COM, STORE_FM.COM, STORE_CON.COM, or STORE_FIP.COM respectively. After executing the appropriate Datatrieve command file, the procedure FIPM_STORE.COM redisplay the store menu. The user may elect to continue storing data in any of the displayed domains or may return to the main menu procedure by selecting line number 7. The program flow for storing FIPM data is shown in Figure 9. A listing of the DCL command procedure FIPM_STORE.COM is included in Appendix D.

If the user selects the modify FIPM data option, the DCL procedure FIPM_MODIFY.COM is called to display the menu shown in Figure 10. A response of 1 through 5 will result in the execution of the Datatrieve command file MODIFY_REF.COM, MODIFY_SYS.COM, MODIFY_MOD.COM, MODIFY_FM.COM, or MODIFY_FIP.COM respectively. The records in domain CONNECTIONS cannot be modified from this menu. After executing the appropriate Datatrieve command file, the procedure FIPM_MODIFY.COM redisplay the modify menu. The user may elect to continue modifying data in any of the displayed domains or may return to the main menu procedure by selecting line number 6. The program flow for modifying FIPM data is shown in Figure 11. A listing of the DCL command procedure FIPM_MODIFY.COM is included in Appendix D.

=====

FAILURE INFORMATION PROPAGATION MODEL

=====

STORE MENU

1. Domain REFERENCES
2. Domain SYSTEMS
3. Domain MODULES
4. Domain FAILUREMODES
5. Domain CONNECTIONS
6. Domain PROPAGATIONS
7. Exit to MAIN MENU

Please enter LINE NUMBER:

FIGURE 8. MENU FOR STORING FIPM DATA BASE RECORDS

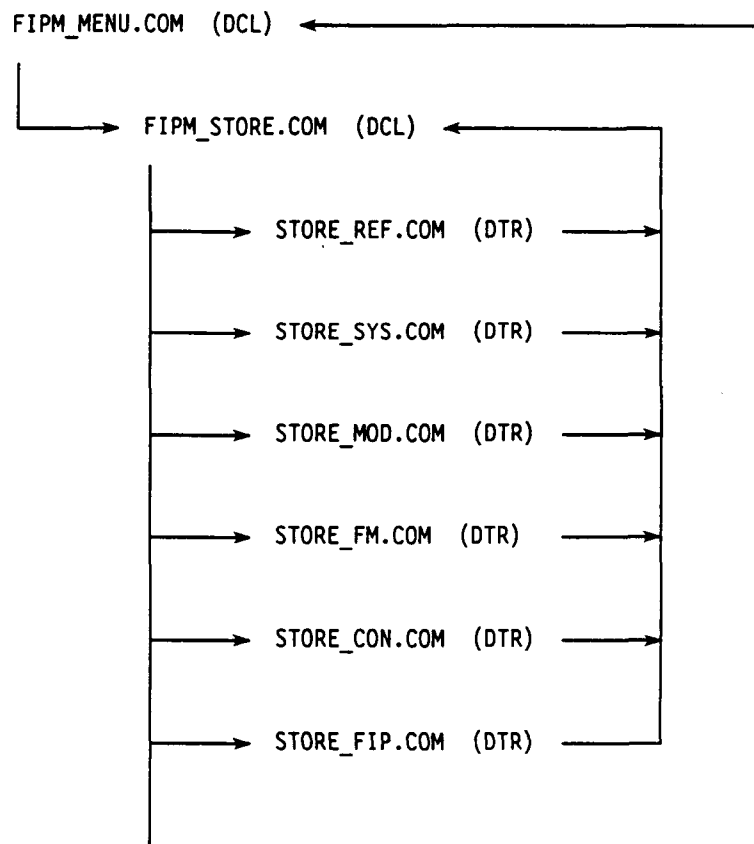


FIGURE 9. PROGRAM FLOW FOR STORING FIPM DATA


```
=====
FAILURE INFORMATION PROPAGATION MODEL
=====
```

MODIFY MENU

1. Domain REFERENCES
2. Domain SYSTEMS
3. Domain MODULES
4. Domain FAILUREMODES
5. Domain PROPAGATIONS
6. Exit to MAIN MENU

Please enter LINE NUMBER:

FIGURE 10. MENU FOR MODIFYING FIPM DATA BASE RECORDS

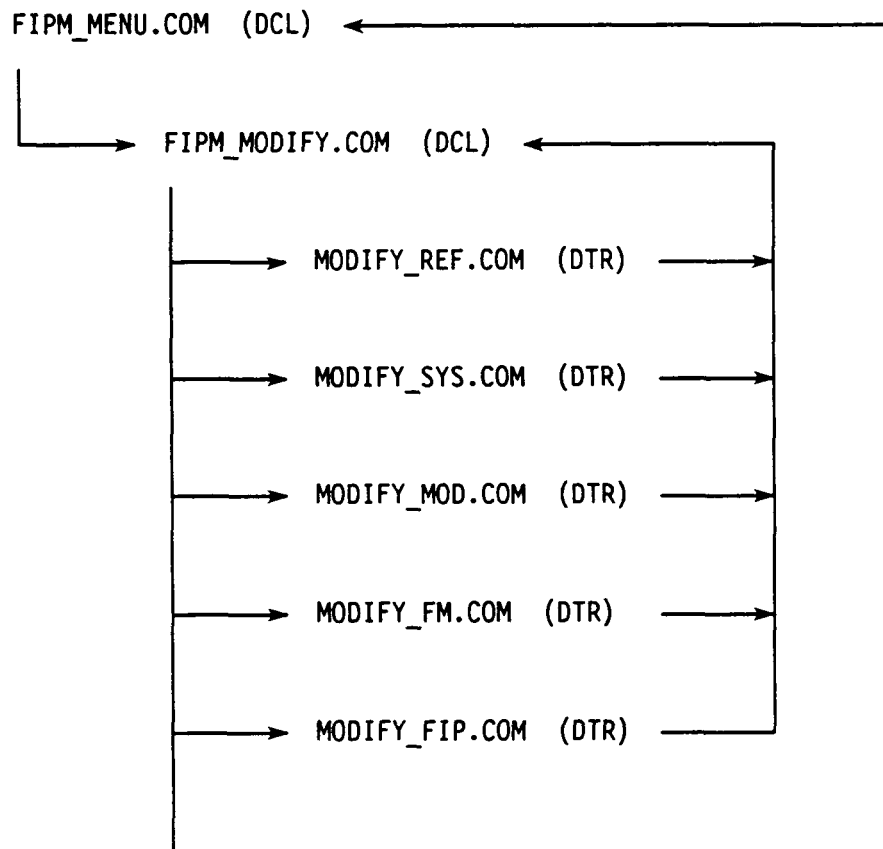


FIGURE 11. PROGRAM FLOW FOR MODIFYING FIPM DATA

If the user selects the list FIPM data option, the DCL procedure FIPM_LIST.COM is called to display the menu shown in Figure 12. A response of 1 through 6 will result in the execution of the Datatrieve command file LIST_REF_1.COM, LIST_SYS_1.COM, LIST_MOD_1.COM, LIST_FM_1.COM, LIST_CON_1.COM, or LIST_FIP_1.COM respectively. After executing the appropriate Datatrieve command file, the procedure FIPM_LIST.COM requests a yes or no response to list the records in the domain. A response of yes results in a batch job being submitted to generate the listing. The procedure then loops back to the list menu. A no response causes the immediate redisplay of the list menu. The user may elect to continue listing data for any of the displayed domains or may return to the main menu by selecting line number 7. The program flow for listing FIPM data is shown in Figure 13. A listing of the DCL command procedure FIPM_LIST.COM is included in Appendix D.

```
=====
                        FAILURE INFORMATION PROPAGATION MODEL
=====
```

```
LIST MENU
```

1. Domain REFERENCES
2. Domain SYSTEMS
3. Domain MODULES
4. Domain FAILUREMODES
5. Domain CONNECTIONS
6. Domain PROPAGATIONS
7. Exit to MAIN MENU

Please enter LINE NUMBER:

FIGURE 12. MENU FOR LISTING FIPM DATA BASE RECORDS

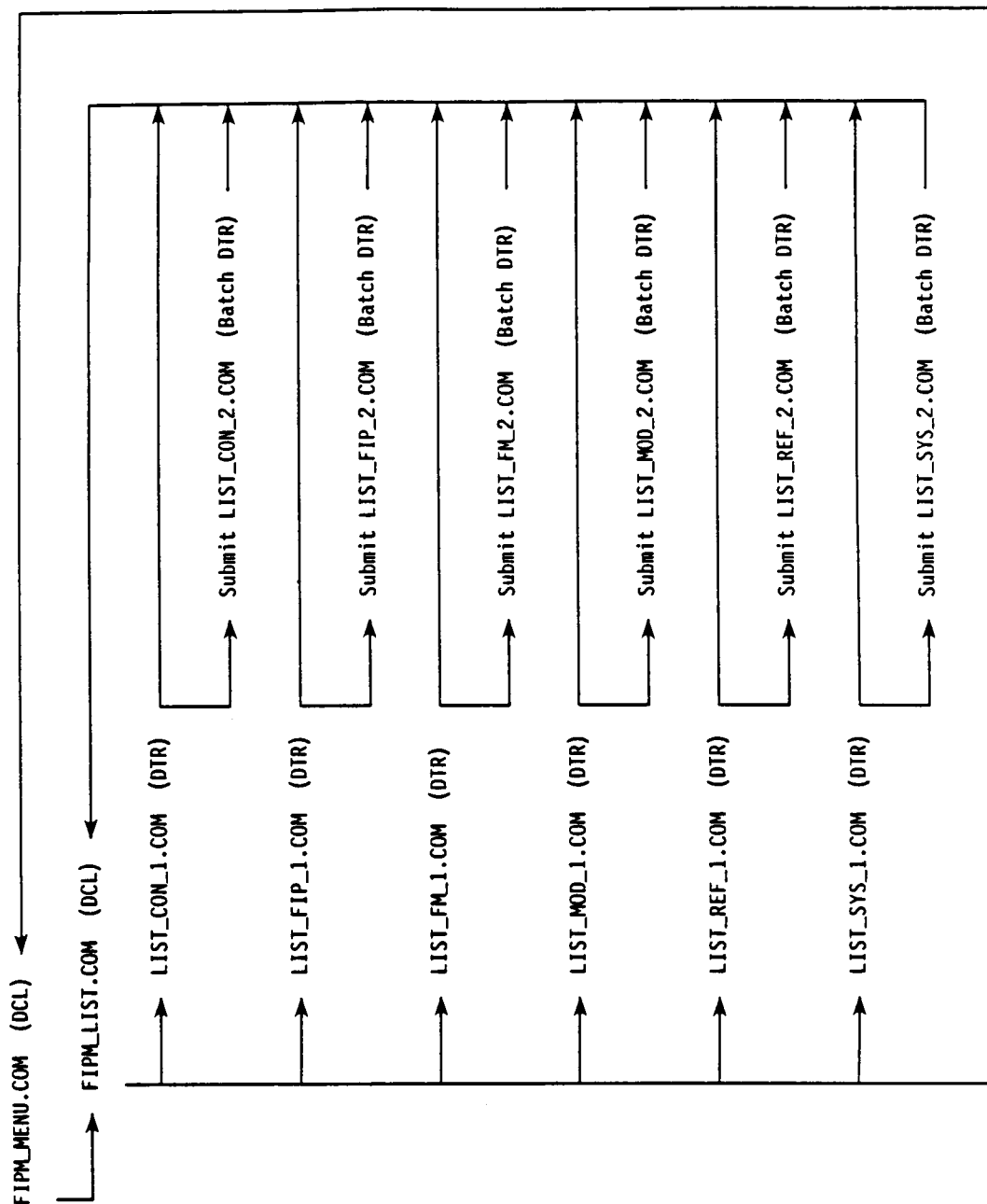


FIGURE 13. PROGRAM FLOW FOR LISTING FIPM DATA

Datatrieve Command Files, Procedures, and Tables

The actual storage, modification, and listing of FIPM information is performed using Datatrieve command files and procedures. Datatrieve command files are VAX system-level files which contain a series of Datatrieve commands and statements. These files are invoked from inside Datatrieve to perform the desired operations. Datatrieve procedures also contain a series of Datatrieve commands and statements. However, these procedures are stored in the VAX Common Data Dictionary (CDD). The CDD is used by Datatrieve to store and access the various elements associated with the data base.

The Datatrieve command files and procedures developed for the FIPM data base use Datatrieve tables to validate and supplement information being entered. An example of a Datatrieve table is shown in Figure 14. These tables are used to validate data by accepting only values which are in the table. They also provide additional data through translation of the value on the left-hand side of the colon into the value on the right-hand side. This latter function was especially useful for creating abbreviations to represent key FIPM data. The overall record size was reduced by storing the abbreviation rather than the entire value.

```

DEFINE TABLE REFERENCE_ABBREV_TABLE
!
"AEROJET"      : "AJ"
"BATTELLE"     : "BA"
"MARTIN MARIETTA" : "MM"
"NASA HDQ"     : "NH"
"NASA MSFC"    : "NM"
"PRATT & WHITNEY" : "PW"
"ROCKETDYNE"   : "RD"
!
END_TABLE

```

FIGURE 14. SAMPLE DATATRIEVE TABLE

The Datatrieve command files, procedures, and tables used to store FIPM information are shown in Figure 15. The command files STORE_REF.COM, STORE_SYS.COM, STORE_MOD.COM, STORE_FM.COM, STORE_CON.COM, and STORE_FIP.COM are executed by the DCL command procedure FIPM_STORE.COM (store menu). Each of these Datatrieve command files opens a log file to document the records being stored in the corresponding domain, prints the current date/time to the log file and then invokes the appropriate Datatrieve procedure(s). After completion of the storage activity, the program flow is returned to the command file where the current date/time is again printed before closing the log file. Execution is then returned to the DCL procedure FIPM_STORE.COM for redisplay of the store menu. The Datatrieve command files, procedures, and tables used to store FIPM data are included in Appendixes E, F, and G respectively.

The Datatrieve command files, procedures, and tables used to modify FIPM information are shown in Figure 16. The command files MODIFY_REF.COM, MODIFY_SYS.COM, MODIFY_MOD.COM, MODIFY_FM.COM, and MODIFY_FIP.COM are executed by the DCL command procedure FIPM_MODIFY.COM (modify menu). Each of these Datatrieve command files opens a log file to document the records being modified in the corresponding domain, prints the current date/time to the log file and then invokes the appropriate Datatrieve procedure(s). After completion of the modification activity, the program flow is returned to the command file where the current date/time is again printed before closing the log file. Execution is then returned to the DCL procedure FIPM_MODIFY.COM for redisplay of the modify menu. The Datatrieve command files, procedures, and tables used to modify FIPM data are included in Appendixes E, F, and G respectively.

Command Files:

STORE_CON.COM
STORE_FIP.COM
STORE_FM.COM
STORE_MOD.COM
STORE_REF.COM
STORE_FIP.COM

Procedures:

BELL
CLRSCRN
CON_STORE
CREATE_PROPAGATIONS_FIP_1
CREATE_PROPAGATIONS_FIP_2
CREATE_PROPAGATIONS_SYS_1
CREATE_PROPAGATIONS_SYS_2
DTR_NULL
FIP_STORE
FIP_STORE_1
FIP_STORE_2
FM_STORE
MOD_STORE
REF_STORE
SYS_STORE

Tables:

ACCOMPLICE_REQUIRED_TABLE
CONNECTION_TABLE
FAILURE_MODE_SUBMODE_TABLE
FMEA_ITEM_NAME_TABLE
FREQ_TIME_UNITS_TABLE
MONTH_TABLE
PARAMETER_TABLE
REFERENCE_ABBREV_TABLE
REFERENCE_SOURCE_TABLE
SIGNAL_TABLE

FIGURE 15. DATATRIEVE COMMAND FILES, PROCEDURES AND TABLES USED TO STORE FIPM DATA

Command Files:

MODIFY_FIP.COM
MODIFY_FM.COM
MODIFY_MOD.COM
MODIFY_REF.COM
MODIFY_SYS.COM

Procedures:

BELL
CLRSCRN
DTR_NULL
FIP_MODIFY
FIP_MODIFY_1
FIP_MODIFY_2
FIP_MODIFY_3
FIP_MODIFY_4
FM_MODIFY
FM_MODIFY_1
MOD_MODIFY
MOD_MODIFY_1
REF_MODIFY
REF_MODIFY_1
SYS_MODIFY
SYS_MODIFY_1
SYS_MODIFY_2

Tables:

FAILURE_MODE_SUBMODE_TABLE
FMEA_ITEM_NAME_TABLE
MONTH_TABLE
NUMBER_TABLE
PARAMETER_TABLE
REFERENCE_SOURCE_TABLE
SIGNAL_TABLE
SIGN_TABLE

FIGURE 16. DATATRIEVE COMMAND FILES, PROCEDURES AND
TABLES USED TO MODIFY FIPM DATA

The Datatrieve command files, procedures, and tables used to list FIPM information are shown in Figure 17. The command files LIST_REF_1.COM, LIST_SYS_1.COM, LIST_MOD_1.COM, LIST_FM_1.COM, LIST_CON_1.COM, and LIST_FIP_1.COM are executed by the DCL command procedure FIPM_LIST.COM (list menu). Each of these Datatrieve command files counts the number of records in the corresponding domain and calculates the number of pages in the list file. This information is printed to the screen and program execution returns to FIPM_LIST.COM. If an actual listing is desired, the appropriate Datatrieve command file LIST_REF_2.COM, LIST_SYS_2.COM, LIST_MOD_2.COM, LIST_FM_2.COM, LIST_CON_2.COM, or LIST_FIP_2.COM is submitted as a batch job to generate the list file. Execution of the DCL procedure FIPM_LIST.COM then continues with redisplay of the list menu. The Datatrieve command files, procedures, and tables used to list FIPM data are included in Appendixes E, F, and G respectively.

A number of Datatrieve procedures were used during the FIPM data base software development to simplify certain functions. As examples, the procedure CREATE_CONNECTIONS executes the file definition command for domain CONNECTIONS and the procedure S132 sets the terminal screen width to 132 characters. In addition to these procedures, the Datatrieve table FMEA_ITEM_PART_NO_TABLE was created to provide the Rocketdyne part numbers (Reference 3, Table 2-1) associated with specific FMEA items. These procedures and table are shown in Figure 18. Listings for the procedures are included in Appendix F and the table is included in Appendix G.

Command Files:

```

LIST_CON_1.COM
LIST_CON_2.COM
LIST_FIP_1.COM
LIST_FIP_2.COM
LIST_FM_1.COM
LIST_FM_2.COM
LIST_MOD_1.COM
LIST_MOD_2.COM
LIST_REF_1.COM
LIST_REF_2.COM
LIST_SYS_1.COM
LIST_SYS_2.COM

```

Procedures:

```

CLRSCRN
DTR_NULL
FIP_COUNT
FIP_COUNT_1
FIP_COUNT_2
FIP_LIST_1
FIP_LIST_2
FIP_LIST_3
FIP_LIST_4

```

FIGURE 17. DATATRIEVE COMMAND FILES AND PROCEDURES USED TO LIST FIPM DATA

Procedures:

```

CREATE_CONNECTIONS
CREATE_CONNECTIONS_FORM
CREATE_FAILUREMODES
CREATE_FAILUREMODES_FORM
CREATE_MODULES
CREATE_MODULES_FORM
CREATE_PROPAGATIONS
CREATE_PROPAGATIONS_FORM
CREATE_REFERENCES
CREATE_REFERENCES_FORM
CREATE_SYSTEMS
CREATE_SYSTEMS_FORM
FIPLOGICALC
FIPLOGICALD
HDR
PRNTOFF
PRNTON
S132
S80

```

Tables:

```

FMEA_ITEM_PART_NO_TABLE

```

FIGURE 18. MISCELLANEOUS DATATRIEVE PROCEDURES AND TABLES USED FOR FIPM

Terminal Data Management System Forms

Two TDMS utilities were used to create and compile terminal forms for use with the FIPM data base. These forms provide the interactive interface between the data base user and the underlying software. The specific utilities used were the Form Definition Utility (FDU) and the Request Definition Utility (RDU). FDU was used to create the screen image, define the video features, assign attributes to the various input fields, establish the field access order, and save the completed form in the Common Data Dictionary (CDD). The FIPM form definitions are included in Appendix H. The RDU was used to create a request library which identifies all of the FIPM forms. The VAX computer file associated with the compiled forms is also specified in the request library definition. Finally, RDU is used to build (compile) the request library and create the library file identified in the definition. The FIPM request library definition is shown in Figure 19.

```

FORM IS CONNECTIONS_STO_FORM;
FORM IS FAILUREMODES_FIN1_FORM;
FORM IS FAILUREMODES_FIN2_FORM;
FORM IS FAILUREMODES_MOD1_FORM;
FORM IS FAILUREMODES_MOD2_FORM;
FORM IS FAILUREMODES_STO1_FORM;
FORM IS FAILUREMODES_STO2_FORM;
FORM IS MODULES_FIN_FORM;
FORM IS MODULES_MOD_FORM;
FORM IS MODULES_STO_FORM;
FORM IS PROPAGATIONS_FIN_FORM;
FORM IS PROPAGATIONS_MOD_FORM;
FORM IS PROPAGATIONS_STO_FORM;
FORM IS REFERENCES_FIN_FORM;
FORM IS REFERENCES_MOD_FORM;
FORM IS REFERENCES_STO_FORM;
FORM IS SYSTEMS_FIN_FORM;
FORM IS SYSTEMS_MOD_FORM;
FORM IS SYSTEMS_STO_FORM;
FILE IS "DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB";
END DEFINITION;

```

FIGURE 19. FIPM REQUEST LIBRARY DEFINITION

(This page intentionally blank)

FIPM DATA BASE TRANSFER

A magnetic tape containing the FIPM data base development software and the high-pressure oxidizer turbopump (HPOTP) data files was mailed to the NASA Marshall Space Flight Center in February 1987. This tape was written using the VAX/VMS Backup Utility and contained multiple copies of 60 files. These files are shown in Figure 20. The file ACTIVATE.COM was a DCL command procedure developed to load and organize all of the required FIPM structure into a newly established VAX username. The 19 files of the type *.DAT are the data files associated with the HPOTP FIPM. The 28 files of the types FIPM_*.COM, LIST_*.COM, LOGIN.COM, MODIFY_*.COM, and STORE_*.COM are the DCL command procedures and Datatrieve command files discussed in the previous section. The three files of the type CDD_FORMS_*.BAK contain the compiled form definitions as extracted from the CDD. The nine files of the type DTR_*.COM contain the Datatrieve domain, record, procedure, and table definitions.

ACTIVATE.COM	CDD_FORMS_1.BAK	CDD_FORMS_2.BAK
CDD_FORMS_3.BAK	CONNECTIONS.DAT	CONNECTIONS_FORM.DAT
DTR_DOMAINS.COM	DTR_PROCS_1.COM	DTR_PROCS_2.COM
DTR_PROCS_3.COM	DTR_PROCS_4.COM	DTR_PROCS_5.COM
DTR_PROCS_6.COM	DTR_RECORDS.COM	DTR_TABLES.COM
FAILUREMODES.DAT	FAILUREMODES_FORM.DAT	FIPM_LIST.COM
FIPM_MENU.COM	FIPM_MODIFY.COM	FIPM_STORE.COM
LIST_CON_1.COM	LIST_CON_2.COM	LIST_FIP_1.COM
LIST_FIP_2.COM	LIST_FM_1.COM	LIST_FM_2.COM
LIST_MOD_1.COM	LIST_MOD_2.COM	LIST_REF_1.COM
LIST_REF_2.COM	LIST_SYS_1.COM	LIST_SYS_2.COM
LOGIN.COM	MODIFY_FIP.COM	MODIFY_FM.COM
MODIFY_MOD.COM	MODIFY_REF.COM	MODIFY_SYS.COM
MODULES.DAT	MODULES_FORM.DAT	PROPAGATIONS_A150.DAT
PROPAGATIONS_A200.DAT	PROPAGATIONS_A600.DAT	PROPAGATIONS_A700.DAT
PROPAGATIONS_B400.DAT	PROPAGATIONS_B800.DAT	PROPAGATIONS_C200.DAT
PROPAGATIONS_FORM.DAT	PROPAGATIONS_Z910.DAT	REFERENCES.DAT
REFERENCES_FORM.DAT	STORE_CON.COM	STORE_FIP.COM
STORE_FM.COM	STORE_MOD.COM	STORE_REF.COM
STORE_SYS.COM	SYSTEMS.DAT	SYSTEMS_FORM.DAT

Total of 60 files.

FIGURE 20. VAX/VMS FILES USED TO TRANSFER FIPM DATA BASE

The procedure `ACTIVATE.COM` created the necessary VAX/VMS directory structure, created a Datatrieve dictionary, loaded all of the Datatrieve elements (domains, records, procedures, and tables), defined a TDMS request library and built the TDMS request library file. A listing of the file `ACTIVATE.COM` is included in Appendix D. The resulting VAX directory structure is shown in Figure 21. The top-level directory, `[BCDSSME2]`, contains the other directory files, the Datatrieve dictionary file, and two DCL command procedures. The files contained in this directory are shown in Figure 22. The directory `[BCDSSME2.DATA]` contains the FIPM data files as shown in Figure 23. The directory `[BCDSSME2.DTR]` is used as a holding area for the command files containing the Datatrieve domain, record, procedure, and table definitions. The files in this directory are shown in Figure 24. The directory `[BCDSSME2.FIPM]` contains the DCL command procedures and Datatrieve command files which display the FIPM menus and interact with the data base. These files are shown in Figure 25. The directory `[BCDSSME2.FORMS]` contains the compiled form definition files and the request library file as shown in Figure 26. The directories `[BCDSSME2.LISTS]`, `[BCDSSME2.LOGS]`, and `[BCDSSME2.MISC]` are initially empty. Any FIPM listing files generated by the FIPM software will be written to the `[BCDSSME2.LISTS]` directory. The log files which are created as FIPM records are stored or modified are written to the directory `[BCDSSME2.LOGS]`. The final directory, `[BCDSSME2.MISC]`, was included for miscellaneous files which may be created by the user.

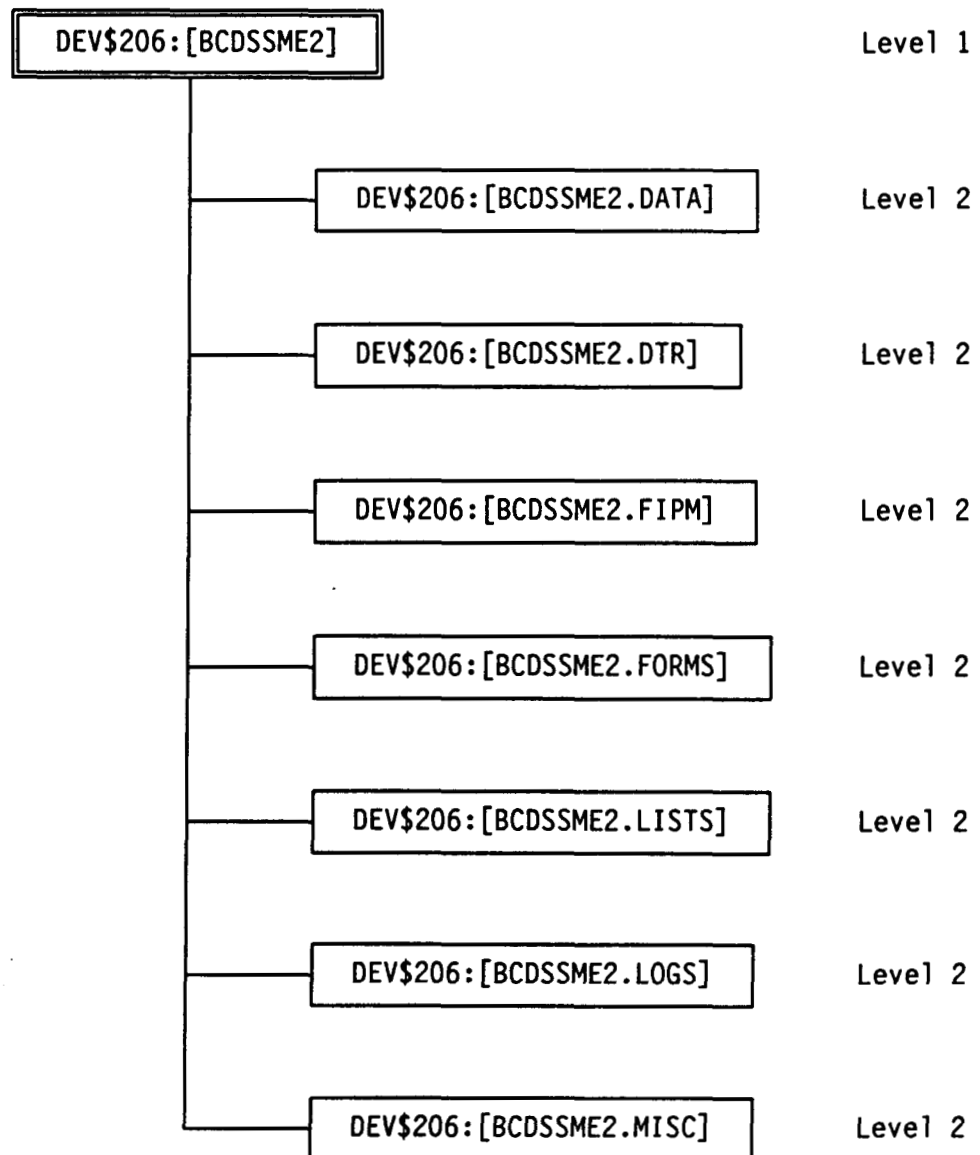


FIGURE 21. FIPM DIRECTORY STRUCTURE

ACTIVATE.COM
DATA.DIR
DTR.DIR
FIPM.DIR
FORMS.DIR
LISTS.DIR
LOGIN.COM
LOGS.DIR
MISC.DIR
SSME.DIC

Total of 10 files.

FIGURE 22. DIRECTORY DEV\$206:[BCDSSME2]

CONNECTIONS.DAT
CONNECTIONS_FORM.DAT
FAILUREMODES.DAT
FAILUREMODES_FORM.DAT
MODULES.DAT
MODULES_FORM.DAT
PROPAGATIONS_A150.DAT
PROPAGATIONS_A200.DAT
PROPAGATIONS_A600.DAT
PROPAGATIONS_A700.DAT
PROPAGATIONS_B400.DAT
PROPAGATIONS_B800.DAT
PROPAGATIONS_C200.DAT
PROPAGATIONS_FORM.DAT
PROPAGATIONS_Z910.DAT
REFERENCES.DAT
REFERENCES_FORM.DAT
SYSTEMS.DAT
SYSTEMS_FORM.DAT

Total of 19 files.

FIGURE 23. DIRECTORY DEV\$206:[BCDSSME2.DATA]

DTR_DOMAINS.COM
DTR_PROCS_1.COM
DTR_PROCS_2.COM
DTR_PROCS_3.COM
DTR_PROCS_4.COM
DTR_PROCS_5.COM
DTR_PROCS_6.COM
DTR_RECORDS.COM
DTR_TABLES.COM

Total of 9 files.

FIGURE 24. DIRECTORY DEV\$206:[BCDSSME2.DTR]

FIPM_LIST.COM	FIPM_MENU.COM
FIPM_MODIFY.COM	FIPM_STORE.COM
LIST_CON_1.COM	LIST_CON_2.COM
LIST_FIP_1.COM	LIST_FIP_2.COM
LIST_FM_1.COM	LIST_FM_2.COM
LIST_MOD_1.COM	LIST_MOD_2.COM
LIST_REF_1.COM	LIST_REF_2.COM
LIST_SYS_1.COM	LIST_SYS_2.COM
MODIFY_FIP.COM	MODIFY_FM.COM
MODIFY_MOD.COM	MODIFY_REF.COM
MODIFY_SYS.COM	STORE_CON.COM
STORE_FIP.COM	STORE_FM.COM
STORE_MOD.COM	STORE_REF.COM
STORE_SYS.COM	

Total of 27 files.

FIGURE 25. DIRECTORY DEV\$206:[BCDSSME2.FIPM]

CDD_FORMS_1.BAK
CDD_FORMS_2.BAK
CDD_FORMS_3.BAK
FORMSLIB.RLB

Total of 4 files.

FIGURE 26. DIRECTORY DEV\$206:[BCDSSME2.FORMS]

(This page intentionally blank)

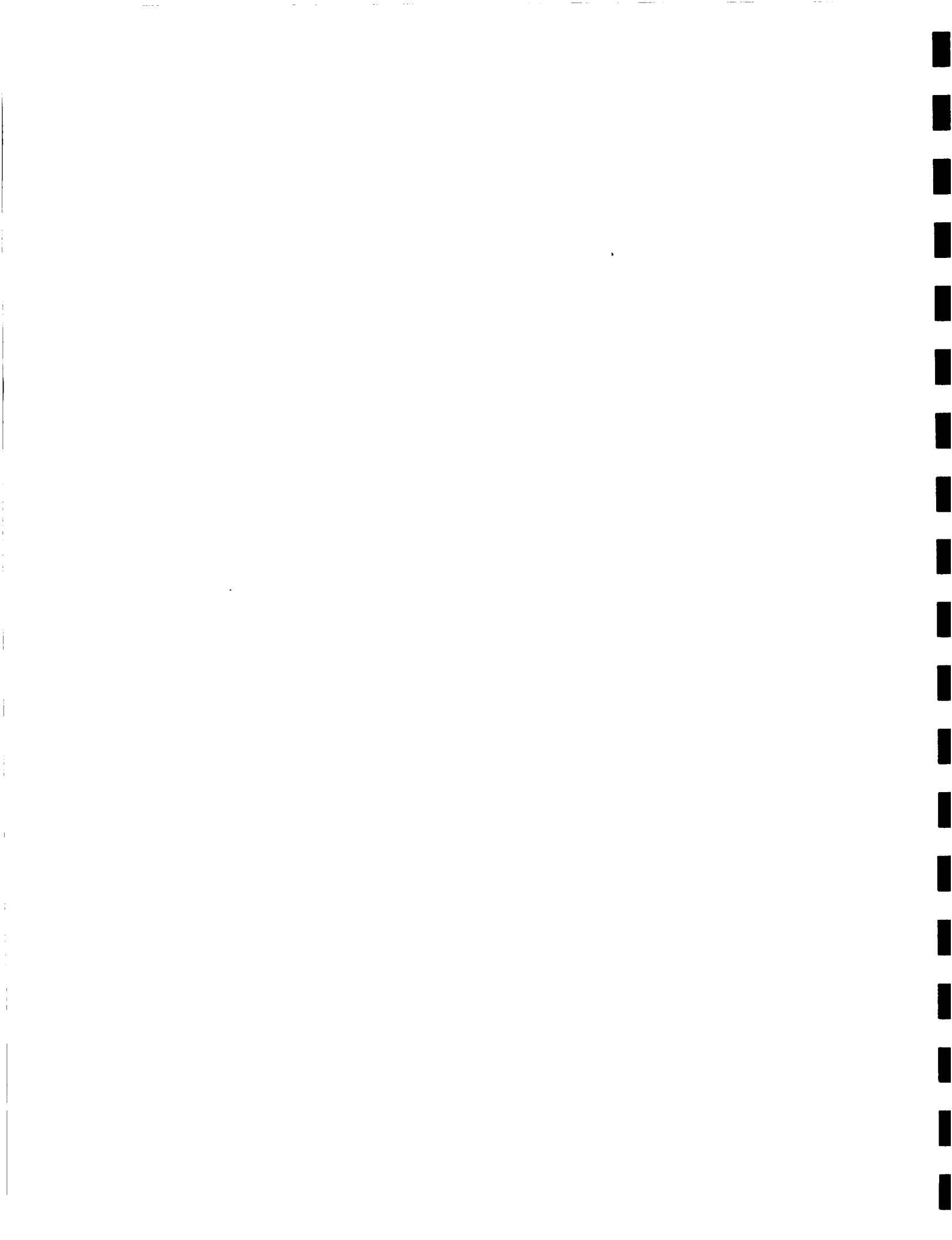
REFERENCES

1. Glover, R. C., Kelley, B. A., and Tischer, A. E., "Studies and Analyses of the Space Shuttle Main Engine, Technical Report Covering SSME Failure Data Review, Diagnostic Survey, and SSME Diagnostic Evaluation", Battelle Memorial Institute, Columbus Division, BCD-SSME-TR-86-1, December 15, 1986, Contract No. NASw-3737.
2. Glover, R. C., Rudy, S. W., and Tischer, A. E., "Studies and Analyses of the Space Shuttle Main Engine, Technical Report on High-Pressure Oxidizer Turbopump Failure Information Propagation Model", Battelle Memorial Institute, Columbus Division, BCD-SSME-TR-87-1, April 20, 1987, Contract No. NASw-3737.
3. "SSME Failure Mode and Effects Analysis and Critical Items List", Rockwell International Corporation, Rocketdyne Division, RSS-8553-9, November 15, 1984, Contract No. NAS8-27980.

PRECEDING PAGE BLANK NOT FILMED

(This page intentionally blank)

APPENDIXES



APPENDIX A

FIPM DOMAIN DEFINITIONS

<u>Domain Definition</u>	<u>Page</u>
CONNECTIONS	A-3
CONNECTIONS_FORM	A-4
FAILUREMODES	A-5
FAILUREMODES_FORM	A-6
MODULES	A-7
MODULES_FORM	A-8
PROPAGATIONS_A150	A-9
PROPAGATIONS_A200	A-10
PROPAGATIONS_A600	A-11
PROPAGATIONS_A700	A-12
PROPAGATIONS_B400	A-13
PROPAGATIONS_B800	A-14
PROPAGATIONS_C200	A-15
PROPAGATIONS_FORM	A-16
PROPAGATIONS_Z910	A-17
REFERENCES	A-18
REFERENCES_FORM	A-19
SYSTEMS	A-20
SYSTEMS_FORM	A-21

A-2

(This page intentionally blank)

Datatrieve Domain CONNECTIONS

DEFINE DOMAIN CONNECTIONS USING CONNECTIONS_REC ON	0001
DEV\$206:[BCDSSME2.DATA]CONNECTIONS.DAT	0002
;	0003

PRECEDING PAGE BLANK NOT FILMED

Datatrieve Domain CONNECTIONS_FORM

```
DEFINE DOMAIN CONNECTIONS_FORM USING CONNECTIONS_REC ON  
DEV$206:[BCDSSME2.DATA]CONNECTIONS_FORM.DAT  
;
```

```
0004  
0005  
0006
```

Datatrieve Domain FAILUREMODES

```
DEFINE DOMAIN FAILUREMODES USING FAILUREMODES_REC ON  
DEV$206:[BCDSSME2.DATA]FAILUREMODES.DAT  
;
```

0007

0008

0009

Datatrieve Domain FAILUREMODES_FORM

DEFINE DOMAIN FAILUREMODES_FORM USING FAILUREMODES_REC ON
DEV\$206:[BCDSSME2.DATA]FAILUREMODES_FORM.DAT
;

0010

0011

0012

Datatrieve Domain MODULES

DEFINE DOMAIN MODULES USING MODULES_REC ON
DEV\$206:[BCDSSME2.DATA]MODULES.DAT
;

0013

0014

0015

Datatrieve Domain MODULES_FORM

```
DEFINE DOMAIN MODULES_FORM USING MODULES_REC ON  
DEV$206:[BCDSSME2.DAT]MODULES_FORM.DAT  
;
```

0016

0017

0018

Datatrieve Domain PROPAGATIONS_A150

FN\$DELETE_LOG("PROPAGATIONS")	0019
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0020
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A150")	0021
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0022
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_A150.DAT")	0023
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0024
;	0025

Datatrieve Domain PROPAGATIONS_A200

FN\$DELETE_LOG("PROPAGATIONS")	0026
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0027
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A200")	0028
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0029
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_A200.DAT")	0030
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0031
;	0032

Datatrieve Domain PROPAGATIONS_A600

FN\$DELETE_LOG("PROPAGATIONS")	0033
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0034
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A600")	0035
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0036
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_A600.DAT")	0037
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0038
;	0039

Datatrieve Domain PROPAGATIONS_A700

FN\$DELETE_LOG("PROPAGATIONS")	0040
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0041
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A700")	0042
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0043
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_A700.DAT")	0044
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0045
;	0046

Datatrieve Domain PROPAGATIONS_B400

FN\$DELETE_LOG("PROPAGATIONS")	0047
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0048
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_B400")	0049
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0050
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_B400.DAT")	0051
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0052
;	0053

Datatrieve Domain PROPAGATIONS_B800

FN\$DELETE_LOG("PROPAGATIONS")	0054
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0055
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_B800")	0056
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0057
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_B800.DAT")	0058
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0059
;	0060

Datatrieve Domain PROPAGATIONS_C200

FN\$DELETE_LOG("PROPAGATIONS")	0061
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0062
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_C200")	0063
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0064
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_C200.DAT")	0065
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0066
;	0067

Datatrieve Domain PROPAGATIONS_FORM

```
DEFINE DOMAIN PROPAGATIONS_FORM USING PROPAGATIONS_REC ON  
DEV$206:[BCDSSME2.DATA]PROPAGATIONS_FORM.DAT  
;
```

0068

0069

0070

Datatrieve Domain PROPAGATIONS_Z910

FN\$DELETE_LOG("PROPAGATIONS")	0071
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0072
FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_Z910")	0073
FN\$CREATE_LOG("PROPAGATIONS_FILE",	0074
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_Z910.DAT")	0075
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE	0076
;	0077

Datatrieve Domain REFERENCES

DEFINE DOMAIN REFERENCES USING REFERENCES_REC ON
DEV\$206:[BCDSSME2.DATA]REFERENCES.DAT
;

0078

0079

0080

Datatrieve Domain REFERENCES_FORM

DEFINE DOMAIN REFERENCES_FORM USING REFERENCES_REC ON
DEV\$206:[BCDSSME2.DATA]REFERENCES_FORM.DAT
;

0081
0082
0083

Datatrieve Domain SYSTEMS

DEFINE DOMAIN SYSTEMS USING SYSTEMS_REC ON
DEV\$206:[BCDSSME2.DATA]SYSTEMS.DAT
;

0084
0085
0086

Datatrieve Domain SYSTEMS_FORM

```
DEFINE DOMAIN SYSTEMS_FORM USING SYSTEMS_REC ON  
DEV$206:[BCDSSME2.DAT]SYSTEMS_FORM.DAT  
;
```

0087

0088

0089

(This page intentionally blank)

APPENDIX B

FIPM RECORD DEFINITIONS

<u>Record Definition</u>	<u>Page</u>
CONNECTIONS_REC	B-3
FAILUREMODES_REC	B-5
MODULES_REC	B-7
PROPAGATIONS_REC	B-9
REFERENCES_REC	B-13
SYSTEMS_REC	B-15

B-2

(This page intentionally blank)

Datatrieve Record CONNECTIONS_REC

DEFINE RECORD CONNECTIONS_REC USING			0001
!			0002
!			0003
01 CONNECTIONS_REC.			0004
!			0005
05 DATE_CREATED	USAGE	IS DATE	0006
	QUERY_NAME	IS DCREATED	0007
	EDIT_STRING	IS X(23).	0008
!			0009
05 CODE_NUMBER	PICTURE	IS X(21)	0010
	QUERY_NAME	IS CODENO.	0011
!			0012
05 CODE_NUMBER_PARTS REDEFINES			0013
CODE_NUMBER.			0014
!			0015
10 SYSTEM_MODULE_A	PICTURE	IS X(8)	0016
	QUERY_NAME	IS SYSMODA.	0017
!			0018
10 SYSTEM_MODULE_A_PARTS REDEFINES			0019
SYSTEM_MODULE_A.			0020
!			0021
15 SYSTEM_A	PICTURE	IS X(4)	0022
	QUERY_NAME	IS SYSA.	0023
!			0024
15 MODULE_A	PICTURE	IS 9(4)	0025
	QUERY_NAME	IS MODA.	0026
!			0027
10 CONNECTION	PICTURE	IS X(4)	0028
	QUERY_NAME	IS CN.	0029
!			0030
10 CONNECTION_PARTS REDEFINES			0031
CONNECTION.			0032
!			0033
15 CONNECTION_TYPE	PICTURE	IS X(2)	0034
	QUERY_NAME	IS CNTYPE.	0035
!			0036
15 CONNECTION_QUALIFIER	PICTURE	IS X(2)	0037
	QUERY_NAME	IS CNQUAL.	0038
!			0039
10 UNANTICIPATED_CONNECTION	PICTURE	IS X(1)	0040
	QUERY_NAME	IS UA.	0041
!			0042
10 SYSTEM_MODULE_B	PICTURE	IS X(8)	0043
	QUERY_NAME	IS SYSMODB.	0044
!			0045
10 SYSTEM_MODULE_B_PARTS REDEFINES			0046
SYSTEM_MODULE_B.			0047
!			0048
15 SYSTEM_B	PICTURE	IS X(4)	0049
	QUERY_NAME	IS SYSB.	0050
!			0051

PRECEDING PAGE BLANK NOT FILMED

Datatrieve Record CONNECTIONS_REC (cont.)

15	MODULE_B	PICTURE	IS 9(4)	0052
		QUERY_NAME	IS MODB.	0053
!				0054
05	DATE_LAST_MODIFIED	USAGE	IS DATE	0055
		QUERY_NAME	IS DLASTMOD	0056
		EDIT_STRING	IS X(23).	0057
!				0058
05	MODIFYING_PROCEDURE	PICTURE	IS X(20)	0059
		QUERY_NAME	IS MODPROC.	0060
!				0061
05	FILLER	PICTURE	IS X(20).	0062
!				0063
;				0064

Datatrieve Record FAILUREMODES_REC

DEFINE RECORD FAILUREMODES_REC USING		0065
!		0066
!		0067
01 FAILUREMODES_REC.		0068
!		0069
05 DATE_CREATED	USAGE IS DATE	0070
	QUERY_NAME IS DCREATED	0071
	EDIT_STRING IS X(23).	0072
!		0073
05 FMCODE	PICTURE IS X(20).	0074
!		0075
05 FMCODE_PARTS REDEFINES FMCODE.		0076
!		0077
		0078
10 SOURCE_SYSTEM_MODULE	PICTURE IS X(8)	0079
	QUERY_NAME IS SSYSMOD.	0080
!		0081
10 SOURCE_SYSTEM_MODULE_PARTS REDEFINES SOURCE_SYSTEM_MODULE.		0082
!		0083
		0084
15 SOURCE_SYSTEM	PICTURE IS X(4)	0085
	QUERY_NAME IS SSYS.	0086
!		0087
15 SOURCE_MODULE	PICTURE IS 9(4)	0088
	QUERY_NAME IS SMOD.	0089
!		0090
10 FAILURE_MODE_SUBMODE	PICTURE IS X(4)	0091
	QUERY_NAME IS FMSUBM.	0092
!		0093
10 FAILURE_MODE_SUBMODE_PARTS REDEFINES FAILURE_MODE_SUBMODE.		0094
!		0095
		0096
15 FAILURE_MODE	PICTURE IS X(2)	0097
	QUERY_NAME IS FM.	0098
!		0099
15 FAILURE_SUBMODE	PICTURE IS X(2)	0100
	QUERY_NAME IS FSUBM.	0101
!		0102
10 ACCOMPLICE_SYSTEM_MODULE	PICTURE IS X(8)	0103
	QUERY_NAME IS ACCSYSMOD.	0104
!		0105
10 ACCOMPLICE_SYSTEM_MODULE_PARTS REDEFINES ACCOMPLICE_SYSTEM_MODULE.		0106
!		0107
		0108
15 ACCOMPLICE_SYSTEM	PICTURE IS X(4)	0109
	QUERY_NAME IS ACCSYS.	0110
!		0111
15 ACCOMPLICE_MODULE	PICTURE IS 9(4)	0112
	QUERY_NAME IS ACCMOD.	0113
!		0114
05 DESCRIPTION	PICTURE IS X(242)	0115

Datatrieve Record FAILUREMODES_REC (cont.)

	QUERY_NAME	IS	DESC	0116
	EDIT_STRING	IS	T(80).	0117
!				0118
05	EFFECTS.			0119
!				0120
10	EFFECT1	PICTURE	IS X(161)	0121
		EDIT_STRING	IS T(80).	0122
!				0123
10	EFFECT2	PICTURE	IS X(161)	0124
		EDIT_STRING	IS T(80).	0125
!				0126
10	EFFECT3	PICTURE	IS X(161)	0127
		EDIT_STRING	IS T(80).	0128
!				0129
10	EFFECT4	PICTURE	IS X(161)	0130
		EDIT_STRING	IS T(80).	0131
!				0132
10	EFFECT5	PICTURE	IS X(161)	0133
		EDIT_STRING	IS T(80).	0134
!				0135
10	EFFECT6	PICTURE	IS X(161)	0136
		EDIT_STRING	IS T(80).	0137
!				0138
05	DATE_LAST_MODIFIED	USAGE	IS DATE	0139
		QUERY_NAME	IS DLASTMOD	0140
		EDIT_STRING	IS X(23).	0141
!				0142
05	MODIFYING_PROCEDURE	PICTURE	IS X(20)	0143
		QUERY_NAME	IS MODPROC.	0144
!				0145
05	FILLER	PICTURE	IS X(100).	0146
!				0147
;				0148

Datatrieve Record MODULES REC

[illegible]

B-8

(This page intentionally blank)

Datatrieve Record PROPAGATIONS_REC

DEFINE RECORD PROPAGATIONS_REC USING		0187
!		0188
!		0189
01 PROPAGATIONS_REC.		0190
!		0191
05 DATE_CREATED	USAGE IS DATE	0192
	QUERY_NAME IS DCREATED	0193
	EDIT_STRING IS X(23).	0194
!		0195
05 FMCODE	PICTURE IS X(20).	0196
!		0197
05 FMCODE_PARTS REDEFINES		0198
FMCODE.		0199
!		0200
10 SOURCE_SYSTEM_MODULE	PICTURE IS X(8)	0201
	QUERY_NAME IS SSYSMOD.	0202
!		0203
10 SOURCE_SYSTEM_MODULE_PARTS REDEFINES		0204
SOURCE_SYSTEM_MODULE.		0205
!		0206
15 SOURCE_SYSTEM	PICTURE IS X(4)	0207
	QUERY_NAME IS SSYS.	0208
!		0209
15 SOURCE_MODULE	PICTURE IS 9(4)	0210
	QUERY_NAME IS SMOD.	0211
!		0212
10 FAILURE_MODE_SUBMODE	PICTURE IS X(4)	0213
	QUERY_NAME IS FMSUBM.	0214
!		0215
10 FAILURE_MODE_SUBMODE_PARTS REDEFINES		0216
FAILURE_MODE_SUBMODE.		0217
!		0218
15 FAILURE_MODE	PICTURE IS X(2)	0219
	QUERY_NAME IS FM.	0220
!		0221
15 FAILURE_SUBMODE	PICTURE IS X(2)	0222
	QUERY_NAME IS FSUBM.	0223
!		0224
10 ACCOMPLICE_SYSTEM_MODULE	PICTURE IS X(8)	0225
	QUERY_NAME IS ACCSYSMOD.	0226
!		0227
10 ACCOMPLICE_SYSTEM_MODULE_PARTS REDEFINES		0228
ACCOMPLICE_SYSTEM_MODULE.		0229
!		0230
15 ACCOMPLICE_SYSTEM	PICTURE IS X(4)	0231
	QUERY_NAME IS ACCSYS.	0232
!		0233
15 ACCOMPLICE_MODULE	PICTURE IS 9(4)	0234
	QUERY_NAME IS ACCMOD.	0235
!		0236
05 CODE_NUMBER	PICTURE IS X(21)	0237

Datatrieve Record PROPAGATIONS_REC (cont.)

	QUERY_NAME IS CODENO.	0238
!		0239
05	CODE_NUMBER_PARTS REDEFINES	0240
	CODE_NUMBER.	0241
!		0242
10	SYSTEM_MODULE_A	0243
	PICTURE IS X(8)	
	QUERY_NAME IS SYSMODA.	0244
!		0245
10	SYSTEM_MODULE_A_PARTS REDEFINES	0246
	SYSTEM_MODULE_A.	0247
!		0248
15	SYSTEM_A	0249
	PICTURE IS X(4)	
	QUERY_NAME IS SYSA.	0250
!		0251
15	MODULE_A	0252
	PICTURE IS 9(4)	
	QUERY_NAME IS MODA.	0253
!		0254
10	CONNECTION	0255
	PICTURE IS X(4)	
	QUERY_NAME IS CN.	0256
!		0257
10	CONNECTION_PARTS REDEFINES	0258
	CONNECTION.	0259
!		0260
15	CONNECTION_TYPE	0261
	PICTURE IS X(2)	
	QUERY_NAME IS CNTYPE.	0262
!		0263
15	CONNECTION_QUALIFIER	0264
	PICTURE IS X(2)	
	QUERY_NAME IS CNQUAL.	0265
!		0266
10	UNANTICIPATED_CONNECTION	0267
	PICTURE IS X(1)	
	QUERY_NAME IS UA.	0268
!		0269
10	SYSTEM_MODULE_B	0270
	PICTURE IS X(8)	
	QUERY_NAME IS SYSMODB.	0271
!		0272
10	SYSTEM_MODULE_B_PARTS REDEFINES	0273
	SYSTEM_MODULE_B.	0274
!		0275
15	SYSTEM_B	0276
	PICTURE IS X(4)	
	QUERY_NAME IS SYSB.	0277
!		0278
15	MODULE_B	0279
	PICTURE IS 9(4)	
	QUERY_NAME IS MODB.	0280
!		0281
05	SIGNAL_DESCRIPTION.	0282
!		0283
10	RAW_SIGNAL.	0284
!		0285
15	SIGNAL_TYPE	0286
	PICTURE IS X(20)	
	QUERY_NAME IS SIG.	0287
!		0288

Datatrieve Record PROPAGATIONS_REC (cont.)

15	SIGNAL_UNITS	PICTURE	IS X(25)	0289
		QUERY_NAME	IS SIGUNIT.	0290
!				0291
15	DIMENSIONS	PICTURE	IS 9(1)	0292
		QUERY_NAME	IS DIM.	0293
!				0294
15	SIGNAL_QUALITY	PICTURE	IS 9(1)	0295
		QUERY_NAME	IS SIGQUAL.	0296
!				0297
15	FREQUENCY_TIME.			0298
!				0299
	20	MAX_FREQ_OR_TIME	PICTURE IS S9(2)	0300
			QUERY_NAME IS MAXFT	0301
!			EDIT_STRING IS -Z9.	0302
				0303
	20	MIN_FREQ_OR_TIME	PICTURE IS S9(2)	0304
			QUERY_NAME IS MINFT	0305
!			EDIT_STRING IS -Z9.	0306
				0307
	20	FT_UNITS	PICTURE IS X(25)	0308
!			QUERY_NAME IS FTUNIT.	0309
				0310
10	SYMPTOM_ELEMENT.			0311
!				0312
	15	SENSITIVE_PARAMETER.		0313
!				0314
	20	PARAMETER	PICTURE IS X(20)	0315
!			QUERY_NAME IS PAR.	0316
				0317
	20	PARAMETER_UNITS	PICTURE IS X(25)	0318
!			QUERY_NAME IS PARUNIT.	0319
				0320
	15	SYMPTOM_DURATION	PICTURE IS S9(2)	0321
!			QUERY_NAME IS SYMDUR	0322
			EDIT_STRING IS -Z9.	0323
				0324
	15	PERIOD_OF_ONSET	PICTURE IS S9(2)	0325
!			QUERY_NAME IS ONSET	0326
			EDIT_STRING IS -Z9.	0327
				0328
	15	INDICATES_FAILURE	PICTURE IS X(1)	0329
!			QUERY_NAME IS INDFAIL.	0330
				0331
10	COMMENTS.			0332
!				0333
	15	COMMENT1	PICTURE IS X(80).	0334
!				0335
	15	COMMENT2	PICTURE IS X(80).	0336
!				0337
	15	COMMENT3	PICTURE IS X(80).	0338
!				0339

Datatrieve Record PROPAGATIONS_REC (cont.)

05	DATE_LAST_MODIFIED	USAGE	IS DATE	0340
		QUERY_NAME	IS DLASTMOD	0341
		EDIT_STRING	IS X(23).	0342
!				0343
05	MODIFYING_PROCEDURE	PICTURE	IS X(20)	0344
		QUERY_NAME	IS MODPROC.	0345
!				0346
05	FILLER	PICTURE	IS X(30).	0347
!				0348
;				0349

Datatrieve Record REFERENCES_REC

DEFINE RECORD REFERENCES_REC USING		0350
!		0351
!		0352
01 REFERENCES_REC.		0353
!		0354
05 DATE_CREATED	USAGE IS DATE	0355
	QUERY_NAME IS DCREATED	0356
	EDIT_STRING IS X(23).	0357
!		0358
05 REFERENCE_NUMBER	PICTURE IS X(5)	0359
	QUERY_NAME IS REFNO.	0360
!		0361
05 REFERENCE_NUMBER_PARTS REDEFINES		0362
REFERENCE_NUMBER.		0363
!		0364
10 SOURCE_ABBREVIATION	PICTURE IS X(2)	0365
	QUERY_NAME IS SABBREV.	0366
!		0367
10 SEQUENCE_NUMBER	PICTURE IS 9(3)	0368
	QUERY_NAME IS SEQNO.	0369
!		0370
05 AUTHORS.		0371
!		0372
10 AUTHOR1	PICTURE IS X(25).	0373
!		0374
10 AUTHOR2	PICTURE IS X(25).	0375
!		0376
10 AUTHOR3	PICTURE IS X(25).	0377
!		0378
10 AUTHOR4	PICTURE IS X(25).	0379
!		0380
05 DOCUMENT_TITLE	PICTURE IS X(161)	0381
	QUERY_NAME IS TITLE	0382
	EDIT_STRING IS T(80).	0383
!		0384
05 DOCUMENT_SOURCE	PICTURE IS X(30)	0385
	QUERY_NAME IS SOURCE.	0386
!		0387
05 DOCUMENT_NUMBER	PICTURE IS X(30)	0388
	QUERY_NAME IS DOCNO.	0389
!		0390
05 DOCUMENT_DATE	PICTURE IS X(11)	0391
	QUERY_NAME IS DOCDATE.	0392
!		0393
05 CONTRACT_NUMBER	PICTURE IS X(20)	0394
	QUERY_NAME IS CONTNO.	0395
!		0396
05 DATE_LAST_MODIFIED	USAGE IS DATE	0397
	QUERY_NAME IS DLASTMOD	0398
	EDIT_STRING IS X(23).	0399
!		0400

Datatrieve Record REFERENCES_REC (cont.)

05	MODIFYING_PROCEDURE	PICTURE	IS X(20)	0401
		QUERY_NAME	IS MODPROC.	0402
!				0403
05	FILLER	PICTURE	IS X(40).	0404
;				0405

Datatrieve Record SYSTEMS_REC

DEFINE RECORD SYSTEMS_REC USING		0406
!		0407
!		0408
01 SYSTEMS_REC.		0409
!		0410
05 DATE_CREATED	USAGE IS DATE	0411
	QUERY_NAME IS DCREATED	0412
	EDIT_STRING IS X(23).	0413
!		0414
05 SYSTEM	PICTURE IS X(4)	0415
	QUERY_NAME IS SYS.	0416
!		0417
05 SYSTEM_NAME	PICTURE IS X(80)	0418
	QUERY_NAME IS SYSNAME.	0419
!		0420
05 FMEA_ITEMS.		0421
!		0422
10 ITEM1	PICTURE IS X(4).	0423
!		0424
10 ITEM2	PICTURE IS X(4).	0425
!		0426
10 ITEM3	PICTURE IS X(4).	0427
!		0428
10 ITEM4	PICTURE IS X(4).	0429
!		0430
10 ITEM5	PICTURE IS X(4).	0431
!		0432
10 ITEM6	PICTURE IS X(4).	0433
!		0434
10 ITEM7	PICTURE IS X(4).	0435
!		0436
10 ITEM8	PICTURE IS X(4).	0437
!		0438
10 ITEM9	PICTURE IS X(4).	0439
!		0440
10 ITEM10	PICTURE IS X(4).	0441
!		0442
10 ITEM11	PICTURE IS X(4).	0443
!		0444
10 ITEM12	PICTURE IS X(4).	0445
!		0446
10 ITEM13	PICTURE IS X(4).	0447
!		0448
10 ITEM14	PICTURE IS X(4).	0449
!		0450
10 ITEM15	PICTURE IS X(4).	0451
!		0452
05 REFERENCES.		0453
!		0454
10 REFERENCE1	PICTURE IS X(5)	0455
	QUERY_NAME IS REF1.	0456

Datatrieve Record SYSTEMS_REC (cont.)

!				0457
	10 REFERENCE2	PICTURE	IS X(5)	0458
		QUERY_NAME	IS REF2.	0459
!				0460
	10 REFERENCE3	PICTURE	IS X(5)	0461
		QUERY_NAME	IS REF3.	0462
!				0463
	10 REFERENCE4	PICTURE	IS X(5)	0464
		QUERY_NAME	IS REF4.	0465
!				0466
	10 REFERENCE5	PICTURE	IS X(5)	0467
		QUERY_NAME	IS REF5.	0468
!				0469
	10 REFERENCE6	PICTURE	IS X(5)	0470
		QUERY_NAME	IS REF6.	0471
!				0472
	10 REFERENCE7	PICTURE	IS X(5)	0473
		QUERY_NAME	IS REF7.	0474
!				0475
	10 REFERENCE8	PICTURE	IS X(5)	0476
		QUERY_NAME	IS REF8.	0477
!				0478
	10 REFERENCE9	PICTURE	IS X(5)	0479
		QUERY_NAME	IS REF9.	0480
!				0481
	10 REFERENCE10	PICTURE	IS X(5)	0482
		QUERY_NAME	IS REF10.	0483
!				0484
	05 PROPAGATIONS_FILE_CREATED	PICTURE	IS X(3)	0485
		QUERY_NAME	IS FIPCREATED.	0486
!				0487
	05 DATE_LAST_MODIFIED	USAGE	IS DATE	0488
		QUERY_NAME	IS DLASTMOD	0489
		EDIT_STRING	IS X(23).	0490
!				0491
	05 MODIFYING_PROCEDURE	PICTURE	IS X(20)	0492
		QUERY_NAME	IS MODPROC.	0493
!				0494
	05 FILLER	PICTURE	IS X(17).	0495
;				0496

APPENDIX C

FIPM FILE DEFINITIONS

<u>Domain File Definition</u>	<u>Page</u>
CONNECTIONS	C-3
CONNECTIONS_FORM	C-4
FAILUREMODES	C-5
FAILUREMODES_FORM	C-6
MODULES	C-7
MODULES_FORM	C-8
PROPAGATIONS_A150	C-9
PROPAGATIONS_A200	C-10
PROPAGATIONS_A600	C-11
PROPAGATIONS_A700	C-12
PROPAGATIONS_B400	C-13
PROPAGATIONS_B800	C-14
PROPAGATIONS_C200	C-15
PROPAGATIONS_FORM	C-16
PROPAGATIONS_Z910	C-17
REFERENCES	C-18
REFERENCES_FORM	C-19
SYSTEMS	C-20
SYSTEMS_FORM	C-21

C-2

(This page intentionally blank)

File Definition for Domain CONNECTIONS

DEFINE FILE FOR CONNECTIONS KEY = DATE_CREATED (DUP),
KEY = CODE_NUMBER

0001
0002

PRECEDING PAGE BLANK NOT FILMED

File Definition for Domain CONNECTIONS_FORM

DEFINE FILE FOR CONNECTIONS_FORM KEY = DATE_CREATED (DUP),
KEY = CODE_NUMBER

0003

0004

File Definition for Domain FAILUREMODES

DEFINE FILE FOR FAILUREMODES KEY = DATE_CREATED (DUP),
KEY = FMCODE

0005
0006

File Definition for Domain FAILUREMODES_FORM

DEFINE FILE FOR FAILUREMODES_FORM KEY = DATE_CREATED (DUP),
KEY = FMCODE

0007
0008

File Definition for Domain MODULES

DEFINE FILE FOR MODULES	KEY = DATE_CREATED	(DUP),	0009
	KEY = SYSTEM_MODULE	,	0010
	KEY = SYSTEM_MODULE_NAME	(DUP)	0011

File Definition for Domain MODULES_FORM

DEFINE FILE FOR MODULES_FORM	KEY = DATE_CREATED	(DUP),	0012
	KEY = SYSTEM_MODULE	,	0013
	KEY = SYSTEM_MODULE_NAME	(DUP)	0014

File Definition for Domain PROPAGATIONS_A150

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A150")	0015
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0016
KEY = FMCODE (DUP),	0017
KEY = CODE_NUMBER (DUP),	0018
KEY = SIGNAL_TYPE (DUP)	0019

File Definition for Domain PROPAGATIONS_A200

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A200")	0020
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0021
KEY = FMCODE (DUP),	0022
KEY = CODE_NUMBER (DUP),	0023
KEY = SIGNAL_TYPE (DUP)	0024

File Definition for Domain PROPAGATIONS_A600

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS A600")	0025
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0026
KEY = FMCODE (DUP),	0027
KEY = CODE_NUMBER (DUP),	0028
KEY = SIGNAL_TYPE (DUP)	0029

File Definition for Domain PROPAGATIONS_A700

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_A700")	0030
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0031
KEY = FMCODE (DUP),	0032
KEY = CODE_NUMBER (DUP),	0033
KEY = SIGNAL_TYPE (DUP)	0034

File Definition for Domain PROPAGATIONS_B400

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_B400")	0035
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0036
KEY = FMCODE (DUP),	0037
KEY = CODE_NUMBER (DUP),	0038
KEY = SIGNAL_TYPE (DUP)	0039

File Definition for Domain PROPAGATIONS_B800

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_B800")	0040
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0041
KEY = FMCODE (DUP),	0042
KEY = CODE_NUMBER (DUP),	0043
KEY = SIGNAL_TYPE (DUP)	0044

File Definition for Domain PROPAGATIONS_C200

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_C200")	0045
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0046
KEY = FMCODE (DUP),	0047
KEY = CODE_NUMBER (DUP),	0048
KEY = SIGNAL_TYPE (DUP)	0049

File Definition for Domain PROPAGATIONS_FORM

DEFINE FILE FOR PROPAGATIONS_FORM	KEY = DATE_CREATED	(DUP),	0050
	KEY = FMCODE	(DUP),	0051
	KEY = CODE_NUMBER	(DUP),	0052
	KEY = SIGNAL_TYPE	(DUP)	0053

File Definition for Domain PROPAGATIONS_Z910

FN\$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_Z910")	0054
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0055
KEY = FMCODE (DUP),	0056
KEY = CODE_NUMBER (DUP),	0057
KEY = SIGNAL_TYPE (DUP)	0058

File Definition for Domain REFERENCES

DEFINE FILE FOR REFERENCES	KEY = DATE_CREATED	(DUP),	0059
	KEY = REFERENCE_NUMBER	,	0060
	KEY = DOCUMENT_TITLE	(DUP),	0061
	KEY = DOCUMENT_SOURCE	(DUP)	0062

File Definition for Domain REFERENCES_FORM

DEFINE FILE FOR REFERENCES_FORM	KEY = DATE_CREATED	(DUP),	0063
	KEY = REFERENCE_NUMBER	,	0064
	KEY = DOCUMENT_TITLE	(DUP),	0065
	KEY = DOCUMENT_SOURCE	(DUP)	0066

File Definition for Domain SYSTEMS

DEFINE FILE FOR SYSTEMS KEY = DATE_CREATED (DUP),	0067
KEY = SYSTEM ,	0068
KEY = SYSTEM_NAME	0069

File Definition for Domain SYSTEMS_FORM

DEFINE FILE FOR SYSTEMS_FORM	KEY = DATE_CREATED (DUP),	0070
	KEY = SYSTEM	0071
	KEY = SYSTEM_NAME	0072

C-22

(This page intentionally blank)

APPENDIX D

FIPM DCL COMMAND PROCEDURES

NOTE

The character string "<ESC>" appears in some of the following procedures. This string represents the ASCII escape character. It is actually a single character which can be observed only while editing the appropriate file using the VAX EDT editor. To insert this character in a file, the EDT special insert (SPECINS) function must be used.

<u>DCL Procedure</u>	<u>Page</u>
ACTIVATE.COM	D-3
FIPM_LIST.COM	D-7
FIPM_MENU.COM	D-11
FIPM_MODIFY.COM	D-13
FIPM_STORE.COM	D-15
LOGIN.COM	D-17

D-2

(This page intentionally blank)

DCL Command Procedure ACTIVATE.COM

```

$! 0001
$!===== 0002
$!===== 0003
$! 0004
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0005
  DEV$206:[BCDSSME2.DATA] 0006
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0007
  DEV$206:[BCDSSME2.DTR] 0008
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0009
  DEV$206:[BCDSSME2.FIPM] 0010
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0011
  DEV$206:[BCDSSME2.FORMS] 0012
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0013
  DEV$206:[BCDSSME2.LISTS] 0014
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0015
  DEV$206:[BCDSSME2.LOGS] 0016
$ CREATE/DIRECTORY/PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) - 0017
  DEV$206:[BCDSSME2.MISC] 0018
$! 0019
$!===== 0020
$!===== 0021
$! 0022
$ RUN SYS$SYSTEM:DMU.EXE 0023
SET DEFAULT CDD$TOP 0024
CREATE/SUBDICTIONARY=DEV$206:[BCDSSME2]SSME.DIC CDD$TOP.SSME 0025
SET DEFAULT CDD$TOP.SSME 0026
RESTORE DEV$206:[BCDSSME2]CDD_FORMS_1.BAK CDD$TOP.SSME 0027
RESTORE DEV$206:[BCDSSME2]CDD_FORMS_2.BAK CDD$TOP.SSME 0028
RESTORE DEV$206:[BCDSSME2]CDD_FORMS_3.BAK CDD$TOP.SSME 0029
EXIT 0030
$ DEFINE CDD$DEFAULT "CDD$TOP.SSME" 0031
$! 0032
$!===== 0033
$!===== 0034
$! 0035
$ RUN SYS$SYSTEM:RDU.EXE 0036
CREATE LIBRARY CDD$TOP.SSME.DTRTDMS 0037
  FORM IS CONNECTIONS_STO_FORM; 0038
  FORM IS FAILUREMODES_FIN1_FORM; 0039
  FORM IS FAILUREMODES_FIN2_FORM; 0040
  FORM IS FAILUREMODES_MOD1_FORM; 0041
  FORM IS FAILUREMODES_MOD2_FORM; 0042
  FORM IS FAILUREMODES_STO1_FORM; 0043
  FORM IS FAILUREMODES_STO2_FORM; 0044
  FORM IS MODULES_FIN_FORM; 0045
  FORM IS MODULES_MOD_FORM; 0046
  FORM IS MODULES_STO_FORM; 0047
  FORM IS PROPAGATIONS_FIN_FORM; 0048
  FORM IS PROPAGATIONS_MOD_FORM; 0049
  FORM IS PROPAGATIONS_STO_FORM; 0050
  FORM IS REFERENCES_FIN_FORM; 0051

```

PRECEDING PAGE BLANK NOT FILMED

DCL Command Procedure ACTIVATE.COM (cont.)

FORM IS REFERENCES_MOD_FORM;	0052
FORM IS REFERENCES_STO_FORM;	0053
FORM IS SYSTEMS_FIN_FORM;	0054
FORM IS SYSTEMS_MOD_FORM;	0055
FORM IS SYSTEMS_STO_FORM;	0056
FILE IS "DEV\$206:[BCDSSME2.FORMS]FORMSLIB.RLB";	0057
END DEFINITION;	0058
EXIT	0059
\$!	0060
\$!=====	0061
\$!=====	0062
\$!	0063
\$ RUN SYS\$SYSTEM:RDU.EXE	0064
BUILD LIBRARY DTRTDMS	0065
EXIT	0066
\$!	0067
\$!=====	0068
\$!=====	0069
\$!	0070
\$ RUN SYS\$SYSTEM:DTR32	0071
@DEV\$206:[BCDSSME2]DTR_DOMAINS.COM	0072
@DEV\$206:[BCDSSME2]DTR_PROCS_1.COM	0073
@DEV\$206:[BCDSSME2]DTR_PROCS_2.COM	0074
@DEV\$206:[BCDSSME2]DTR_PROCS_3.COM	0075
@DEV\$206:[BCDSSME2]DTR_PROCS_4.COM	0076
@DEV\$206:[BCDSSME2]DTR_PROCS_5.COM	0077
@DEV\$206:[BCDSSME2]DTR_PROCS_6.COM	0078
@DEV\$206:[BCDSSME2]DTR_RECORDS.COM	0079
@DEV\$206:[BCDSSME2]DTR_TABLES.COM	0080
EXIT	0081
\$!	0082
\$!=====	0083
\$!=====	0084
\$!	0085
\$ RENAME DEV\$206:[BCDSSME2]*.DAT;* DEV\$206:[BCDSSME2.DATA]*.*;	0086
\$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -	0087
DEV\$206:[BCDSSME2.DATA]*.*;	0088
\$!	0089
\$!=====	0090
\$!=====	0091
\$!	0092
\$ RENAME DEV\$206:[BCDSSME2]DTR*.COM;* DEV\$206:[BCDSSME2.DTR]*.*;	0093
\$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -	0094
DEV\$206:[BCDSSME2.DTR]*.*;	0095
\$!	0096
\$!=====	0097
\$!=====	0098
\$!	0099
\$ RENAME DEV\$206:[BCDSSME2]FIPM*.COM;* DEV\$206:[BCDSSME2.FIPM]*.*;	0100
\$ RENAME DEV\$206:[BCDSSME2]LIST*.COM;* DEV\$206:[BCDSSME2.FIPM]*.*;	0101
\$ RENAME DEV\$206:[BCDSSME2]MODIFY*.COM;* DEV\$206:[BCDSSME2.FIPM]*.*;	0102

DCL Command Procedure ACTIVATE.COM (cont.)

\$ RENAME DEV\$206:[BCDSSME2]STORE*.COM;* DEV\$206:[BCDSSME2.FIPM]*.*;*	0103
\$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -	0104
DEV\$206:[BCDSSME2.FIPM]*.*;*	0105
\$!	0106
\$!=====	0107
\$!=====	0108
\$!	0109
\$ RENAME DEV\$206:[BCDSSME2]*.BAK;* DEV\$206:[BCDSSME2.FORMS]*.*;*	0110
\$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -	0111
DEV\$206:[BCDSSME2.FORMS]*.*;*	0112
\$!	0113
\$!=====	0114
\$!=====	0115
\$!	0116
\$ @DEV\$206:[BCDSSME2]LOGIN.COM	0117
\$!	0118
\$!=====	0119
\$!=====	0120
\$!	0121

D-6

(This page intentionally blank)

DCL Command Procedure FIPM_LIST.COM

```

$ ON CONTROL_Y THEN GOTO LOOP                                0122
$ LOOP:                                                         0123
$   OUTPUT := WRITE SYS$OUTPUT                                0124
$   OUTPUT "<ESC>[2J"                                           0125
$   OUTPUT "<ESC>[1;1H"                                         0126
$   OUTPUT " "                                                 0127
$   OUTPUT "          ====="                                0128
$   OUTPUT " "                                                 0129
$   OUTPUT "          FAILURE INFORMATION PROPAGATION MODEL"  0130
$   OUTPUT " "                                                 0131
$   OUTPUT "          ====="                                0132
$   OUTPUT " "                                                 0133
$   OUTPUT "          LIST MENU"                               0134
$   OUTPUT " "                                                 0135
$   OUTPUT " "                                                 0136
$   OUTPUT "          1. Domain REFERENCES"                    0137
$   OUTPUT "          2. Domain SYSTEMS"                      0138
$   OUTPUT "          3. Domain MODULES"                      0139
$   OUTPUT "          4. Domain FAILUREMODES"                 0140
$   OUTPUT "          5. Domain CONNECTIONS"                  0141
$   OUTPUT "          6. Domain PROPAGATIONS"                 0142
$   OUTPUT "          7. Exit to MAIN MENU"                   0143
$   OUTPUT " "                                                 0144
$   OUTPUT " "                                                 0145
$   INQUIRE NUMBER "          Please enter LINE NUMBER"      0146
$   IF NUMBER .EQS. "1" THEN GOTO REF                          0147
$   IF NUMBER .EQS. "2" THEN GOTO SYS                          0148
$   IF NUMBER .EQS. "3" THEN GOTO MOD                          0149
$   IF NUMBER .EQS. "4" THEN GOTO FM                           0150
$   IF NUMBER .EQS. "5" THEN GOTO CON                          0151
$   IF NUMBER .EQS. "6" THEN GOTO FIP                          0152
$   IF NUMBER .EQS. "7" THEN $ EXIT                            0153
$   GOTO LOOP                                                  0154
$ REF:                                                         0155
$   OUTPUT "<ESC>[2J"                                           0156
$   DEFINE/USER_MODE SYS$INPUT SYS$COMMAND                    0157
$   DTR32 @DEV$206:[BCDSSME2.FIPM]LIST_REF_1.COM              0158
$   REF1:                                                       0159
$   OUTPUT " "                                                 0160
$   OUTPUT " "                                                 0161
$   OUTPUT "Do you wish to list Domain REFERENCES?"           0162
$   OUTPUT " "                                                 0163
$   INQUIRE CONT1 "Enter Y or N"                              0164
$   IF CONT1 .EQS. "N" THEN GOTO LOOP                          0165
$   IF CONT1 .NES. "Y" THEN GOTO REF1                          0166
$   DEFINE/USER_MODE SYS$OUTPUT NL:                            0167
$   SUBMIT/NOPRINTER/LOG_FILE=DEV$206:[BCDSSME2.LOGS] -      0168
$       DEV$206:[BCDSSME2.FIPM]LIST_REF_2.COM                 0169
$   OUTPUT " "                                                 0170
$   OUTPUT " "                                                 0171
$   OUTPUT "The listing will be saved in file"                 0172

```

DCL Command Procedure FIPM_LIST.COM (cont.)

```

$      OUTPUT "DEV$206:[BCDSSME2.LISTS]LIST_REF.LST." 0173
$      OUTPUT "Please note this file name for future reference." 0174
$      OUTPUT " " 0175
$      INQUIRE/NOPUNCTUATION CONT2 - 0176
$          "Enter any character and RETURN to continue ..." 0177
$      GOTO LOOP 0178
$ SYS: 0179
$      OUTPUT "<ESC>[2J" 0180
$      DEFINE/USER_MODE SYS$INPUT SYS$COMMAND 0181
$      DTR32 @DEV$206:[BCDSSME2.FIPM]LIST_SYS_1.COM 0182
$      SYS1: 0183
$          OUTPUT " " 0184
$          OUTPUT " " 0185
$          OUTPUT "Do you wish to list Domain SYSTEMS?" 0186
$          OUTPUT " " 0187
$          INQUIRE CONT1 "Enter Y or N" 0188
$          IF CONT1 .EQS. "N" THEN GOTO LOOP 0189
$          IF CONT1 .NES. "Y" THEN GOTO SYS1 0190
$          DEFINE/USER_MODE SYS$OUTPUT NL: 0191
$          SUBMIT/NOPRINTER/LOG_FILE=DEV$206:[BCDSSME2.LOGS] - 0192
$              DEV$206:[BCDSSME2.FIPM]LIST_SYS_2.COM 0193
$          OUTPUT " " 0194
$          OUTPUT " " 0195
$          OUTPUT "The listing will be saved in file" 0196
$          OUTPUT "DEV$206:[BCDSSME2.LISTS]LIST_SYS.LST." 0197
$          OUTPUT "Please note this file name for future reference." 0198
$          OUTPUT " " 0199
$          INQUIRE/NOPUNCTUATION CONT2 - 0200
$              "Enter any character and RETURN to continue ..." 0201
$      GOTO LOOP 0202
$ MOD: 0203
$      OUTPUT "<ESC>[2J" 0204
$      DEFINE/USER_MODE SYS$INPUT SYS$COMMAND 0205
$      DTR32 @DEV$206:[BCDSSME2.FIPM]LIST_MOD_1.COM 0206
$      MOD1: 0207
$          OUTPUT " " 0208
$          OUTPUT " " 0209
$          OUTPUT "Do you wish to list Domain MODULES?" 0210
$          OUTPUT " " 0211
$          INQUIRE CONT1 "Enter Y or N" 0212
$          IF CONT1 .EQS. "N" THEN GOTO LOOP 0213
$          IF CONT1 .NES. "Y" THEN GOTO MOD1 0214
$          DEFINE/USER_MODE SYS$OUTPUT NL: 0215
$          SUBMIT/NOPRINTER/LOG_FILE=DEV$206:[BCDSSME2.LOGS] - 0216
$              DEV$206:[BCDSSME2.FIPM]LIST_MOD_2.COM 0217
$          OUTPUT " " 0218
$          OUTPUT " " 0219
$          OUTPUT "The listing will be saved in file" 0220
$          OUTPUT "DEV$206:[BCDSSME2.LISTS]LIST_MOD.LST." 0221
$          OUTPUT "Please note this file name for future reference." 0222
$          OUTPUT " " 0223

```

\$	INQUIRE/NOPUNCTUATION CONT2 -	0224
	"Enter any character and RETURN to continue ..."	0225
\$	GOTO LOOP	0226
\$	FM:	0227
\$	OUTPUT "<ESC>[2J"	0228
\$	DEFINE/USER MODE SYS\$INPUT SYS\$COMMAND	0229
\$	DTR32 @DEV\$206:[BCDSSME2.FIPM]LIST_FM_1.COM	0230
\$	FM1:	0231
\$	OUTPUT " "	0232
\$	OUTPUT " "	0233
\$	OUTPUT "Do you wish to list Domain FAILUREMODES?"	0234
\$	OUTPUT " "	0235
\$	INQUIRE CONT1 "Enter Y or N"	0236
\$	IF CONT1 .EQS. "N" THEN GOTO LOOP	0237
\$	IF CONT1 .NES. "Y" THEN GOTO FM1	0238
\$	DEFINE/USER MODE SYS\$OUTPUT NL:	0239
\$	SUBMIT/NOPRINTER/LOG_FILE=DEV\$206:[BCDSSME2.LOGS] -	0240
	DEV\$206:[BCDSSME2.FIPM]LIST_FM_2.COM	0241
\$	OUTPUT " "	0242
\$	OUTPUT " "	0243
\$	OUTPUT "The listing will be saved in file"	0244
\$	OUTPUT "DEV\$206:[BCDSSME2.LISTS]LIST_FM.LST."	0245
\$	OUTPUT "Please note this file name for future reference."	0246
\$	OUTPUT " "	0247
\$	INQUIRE/NOPUNCTUATION CONT2 -	0248
	"Enter any character and RETURN to continue ..."	0249
\$	GOTO LOOP	0250
\$	CON:	0251
\$	OUTPUT "<ESC>[2J"	0252
\$	DEFINE/USER MODE SYS\$INPUT SYS\$COMMAND	0253
\$	DTR32 @DEV\$206:[BCDSSME2.FIPM]LIST_CON_1.COM	0254
\$	CON1:	0255
\$	OUTPUT " "	0256
\$	OUTPUT " "	0257
\$	OUTPUT "Do you wish to list Domain CONNECTIONS?"	0258
\$	OUTPUT " "	0259
\$	INQUIRE CONT1 "Enter Y or N"	0260
\$	IF CONT1 .EQS. "N" THEN GOTO LOOP	0261
\$	IF CONT1 .NES. "Y" THEN GOTO CON1	0262
\$	DEFINE/USER MODE SYS\$OUTPUT NL:	0263
\$	SUBMIT/NOPRINTER/LOG_FILE=DEV\$206:[BCDSSME2.LOGS] -	0264
	DEV\$206:[BCDSSME2.FIPM]LIST_CON_2.COM	0265
\$	OUTPUT " "	0266
\$	OUTPUT " "	0267
\$	OUTPUT "The listing will be saved in file"	0268
\$	OUTPUT "DEV\$206:[BCDSSME2.LISTS]LIST_CON.LST."	0269
\$	OUTPUT "Please note this file name for future reference."	0270
\$	OUTPUT " "	0271
\$	INQUIRE/NOPUNCTUATION CONT2 -	0272
	"Enter any character and RETURN to continue ..."	0273
\$	GOTO LOOP	0274

DCL Command Procedure FIPM_LIST.COM (cont.)

\$ FIP:	0275
\$ OUTPUT "<ESC>[2J"	0276
\$ DEFINE/USER_MODE SYS\$INPUT SYS\$COMMAND	0277
\$ DTR32 @DEV\$206:[BCDSSME2.FIPM]LIST_FIP_1.COM	0278
\$ FIP1:	0279
\$ OUTPUT " "	0280
\$ OUTPUT " "	0281
\$ OUTPUT "Do you wish to list the Failure Information"	0282
\$ OUTPUT "Propagation domains?"	0283
\$ OUTPUT " "	0284
\$ INQUIRE CONT1 "Enter Y or N"	0285
\$ IF CONT1 .EQS. "N" THEN GOTO LOOP	0286
\$ IF CONT1 .NES. "Y" THEN GOTO FIP1	0287
\$ DEFINE/USER_MODE SYS\$OUTPUT NL:	0288
\$ SUBMIT/NOPRINTER/LOG_FILE=DEV\$206:[BCDSSME2.LOGS] -	0289
DEV\$206:[BCDSSME2.FIPM]LIST_FIP_2.COM	0290
\$ OUTPUT " "	0291
\$ OUTPUT " "	0292
\$ OUTPUT "The listing will be saved in file"	0293
\$ OUTPUT "DEV\$206:[BCDSSME2.LISTS]LIST_FIP.LST."	0294
\$ OUTPUT "Please note this file name for future reference."	0295
\$ OUTPUT " "	0296
\$ INQUIRE/NOPUNCTUATION CONT2 -	0297
"Enter any character and RETURN to continue ..."	0298
\$ GOTO LOOP	0299

DCL Command Procedure FIPM_MENU.COM

```

$ ON CONTROL_Y THEN GOTO START                                0300
$ START:                                                         0301
$   SET TERMINAL/ECHO                                           0302
$ LOOP:                                                         0303
$   OUTPUT := WRITE SYSS$OUTPUT                                0304
$   OUTPUT "<ESC>[2J"                                           0305
$   OUTPUT "<ESC>[1;1H"                                         0306
$   OUTPUT " "                                                 0307
$   OUTPUT " "                                                 0308
$   OUTPUT " "                                                 0309
$   OUTPUT " ===== "                                       0310
$   OUTPUT " "                                                 0311
$   OUTPUT "          FAILURE INFORMATION PROPAGATION MODEL" 0312
$   OUTPUT " "                                                 0313
$   OUTPUT " ===== "                                       0314
$   OUTPUT " "                                                 0315
$   OUTPUT " "                                                 0316
$   OUTPUT "          MAIN MENU"                               0317
$   OUTPUT " "                                                 0318
$   OUTPUT "          1. Store FIPM Data"                       0319
$   OUTPUT "          2. Modify FIPM Data"                     0320
$   OUTPUT "          3. List FIPM Data"                       0321
$   OUTPUT "          4. Exit Procedure and Logout"            0322
$   OUTPUT " "                                                 0323
$   OUTPUT " "                                                 0324
$   INQUIRE NUMBER "          Please enter LINE NUMBER"      0325
$   IF NUMBER .EQS. "1" THEN -                                  0326
$     $ @DEV$206:[BCDSSME2.FIPM]FIPM_STORE.COM                0327
$   IF NUMBER .EQS. "2" THEN -                                  0328
$     $ @DEV$206:[BCDSSME2.FIPM]FIPM_MODIFY.COM                0329
$   IF NUMBER .EQS. "3" THEN -                                  0330
$     $ @DEV$206:[BCDSSME2.FIPM]FIPM_LIST.COM                  0331
$   IF NUMBER .EQS. "4" THEN GOTO LGOUT                         0332
$   IF NUMBER .EQS. "***" THEN GOTO SYSTM                       0333
$   GOTO LOOP                                                  0334
$ SYSTM:                                                         0335
$   OUTPUT " "                                                 0336
$   SET TERMINAL/NOECHO                                         0337
$   INQUIRE PASSWORD1 "          Please enter PASSWORD 1"     0338
$   SET TERMINAL/ECHO                                           0339
$   IF PASSWORD1 .NES. "*****" THEN GOTO LOOP                 0340
$   OUTPUT " "                                                 0341
$   SET TERMINAL/NOECHO                                         0342
$   INQUIRE PASSWORD2 "          Please enter PASSWORD 2"     0343
$   SET TERMINAL/ECHO                                           0344
$   IF PASSWORD2 .NES. "*****" THEN GOTO LOOP                 0345
$   OUTPUT "<ESC>[2J"                                           0346
$   OUTPUT "<ESC>[1;1H"                                         0347
$   EXIT                                                         0348
$ LGOUT:                                                         0349
$   SET NOCONTROL = Y                                          0350

```


DCL Command Procedure FIPM_MENU.COM (cont.)

\$ LOGOUT

0351

DCL Command Procedure FIPM_MODIFY.COM (cont.)

\$ FIP:	0403
\$ OUTPUT "<ESC>[2J"	0404
\$ DEFINE/USER_MODE SYS\$INPUT SYS\$COMMAND	0405
\$ DTR32 @DEV\$206:[BCDSSME2.FIPM]MODIFY_FIP.COM	0406
\$ GOTO LOOP	0407

DCL Command Procedure FIPM_STORE.COM

```

$ ON CONTROL_Y THEN GOTO LOOP                                0408
$ LOOP:                                                         0409
$   OUTPUT := WRITE SYSS$OUTPUT                                0410
$   OUTPUT "<ESC>[2J"                                           0411
$   OUTPUT "<ESC>[1;1H"                                         0412
$   OUTPUT " "                                                 0413
$   OUTPUT "===== "                                         0414
$   OUTPUT " "                                                 0415
$   OUTPUT "          FAILURE INFORMATION PROPAGATION MODEL" 0416
$   OUTPUT " "                                                 0417
$   OUTPUT "===== "                                         0418
$   OUTPUT " "                                                 0419
$   OUTPUT " "                                                 0420
$   OUTPUT "          STORE MENU"                             0421
$   OUTPUT " "                                                 0422
$   OUTPUT "          1. Domain REFERENCES"                   0423
$   OUTPUT "          2. Domain SYSTEMS"                     0424
$   OUTPUT "          3. Domain MODULES"                     0425
$   OUTPUT "          4. Domain FAILUREMODES"                0426
$   OUTPUT "          5. Domain CONNECTIONS"                 0427
$   OUTPUT "          6. Domain PROPAGATIONS"                0428
$   OUTPUT "          7. Exit to MAIN MENU"                  0429
$   OUTPUT " "                                                 0430
$   OUTPUT " "                                                 0431
$   INQUIRE NUMBER "          Please enter LINE NUMBER"     0432
$   IF NUMBER .EQS. "1" THEN GOTO REF                         0433
$   IF NUMBER .EQS. "2" THEN GOTO SYS                         0434
$   IF NUMBER .EQS. "3" THEN GOTO MOD                         0435
$   IF NUMBER .EQS. "4" THEN GOTO FM                         0436
$   IF NUMBER .EQS. "5" THEN GOTO CON                         0437
$   IF NUMBER .EQS. "6" THEN GOTO FIP                         0438
$   IF NUMBER .EQS. "7" THEN $ EXIT                           0439
$   GOTO LOOP                                                  0440
$ REF:                                                         0441
$   OUTPUT "<ESC>[2J"                                           0442
$   DEFINE/USER_MODE SYSS$INPUT SYSS$COMMAND                 0443
$   DTR32 @DEV$206:[BCDSSME2.FIPM]STORE_REF.COM              0444
$   GOTO LOOP                                                  0445
$ SYS:                                                         0446
$   OUTPUT "<ESC>[2J"                                           0447
$   DEFINE/USER_MODE SYSS$INPUT SYSS$COMMAND                 0448
$   DTR32 @DEV$206:[BCDSSME2.FIPM]STORE_SYS.COM              0449
$   GOTO LOOP                                                  0450
$ MOD:                                                         0451
$   OUTPUT "<ESC>[2J"                                           0452
$   DEFINE/USER_MODE SYSS$INPUT SYSS$COMMAND                 0453
$   DTR32 @DEV$206:[BCDSSME2.FIPM]STORE_MOD.COM              0454
$   GOTO LOOP                                                  0455
$ FM:                                                         0456
$   OUTPUT "<ESC>[2J"                                           0457
$   DEFINE/USER_MODE SYSS$INPUT SYSS$COMMAND                 0458

```

DCL Command Procedure FIPM_STORE.COM (cont.)

\$	DTR32 @DEV\$206:[BCDSSME2.FIPM]STORE_FM.COM	0459
\$	GOTO LOOP	0460
\$	CON:	0461
\$	OUTPUT "<ESC>[2J"	0462
\$	DEFINE/USER_MODE SYS\$INPUT SYS\$COMMAND	0463
\$	DTR32 @DEV\$206:[BCDSSME2.FIPM]STORE_CON.COM	0464
\$	GOTO LOOP	0465
\$	FIP:	0466
\$	OUTPUT "<ESC>[2J"	0467
\$	DEFINE/USER_MODE SYS\$INPUT SYS\$COMMAND	0468
\$	DTR32 @DEV\$206:[BCDSSME2.FIPM]STORE_FIP.COM	0469
\$	GOTO LOOP	0470

DCL Command Procedure LOGIN.COM

```

$ ON WARNING THEN GOTO PROMPT                                0471
$ ON CONTROL_Y THEN GOTO LGOUT                                0472
$ DEFINE CDD$DEFAULT "CDD$TOP.SSME"                          0473
$ IF F$MODE() .NES. "INTERACTIVE" THEN EXIT                  0474
$ GOTO START                                                  0475
$ LGOUT:                                                       0476
$   SET NOCONTROL = Y                                         0477
$   SET TERMINAL/ECHO                                         0478
$   LOGOUT                                                    0479
$ START:                                                       0480
$   SET TERMINAL/DEVICE TYPE = VT100                          0481
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWED,GROUP,WORLD)/DEFAULT 0482
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -      0483
$     DEV$206:[BCDSSME2]*.*;*                                0484
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -      0485
$     DEV$206:[BCDSSME2.DATA]*.*;*                            0486
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -      0487
$     DEV$206:[BCDSSME2.DTR]*.*;*                             0488
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -      0489
$     DEV$206:[BCDSSME2.FIPM]*.*;*                            0490
$   SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) -      0491
$     DEV$206:[BCDSSME2.FORMS]*.*;*                           0492
$   SET DEFAULT [BCDSSME2.MISC]                                0493
$   FIPM  ::= @DEV$206:[BCDSSME2.FIPM]FIPM_MENU.COM          0494
$   DTR32 ::= $SYS$SYSTEM:DTR32                                0495
$   ON WARNING THEN GOTO DIRECTORY2                           0496
$   SET MESSAGE/NOFACILITY -                                  0497
$     /NOIDENTIFICATION -                                     0498
$     /NOSEVERITY -                                           0499
$     /NOTEXT                                                  0500
$   DIRECTORY/TOTAL/SIZE=ALL DEV$206:[BCDSSME2.LISTS]        0501
$   DIRECTORY2:                                                0502
$     ON WARNING THEN GOTO PROMPT                              0503
$     DIRECTORY/TOTAL/SIZE=ALL DEV$206:[BCDSSME2.LOGS]        0504
$     WRITE SYS$OUTPUT " "                                     0505
$     WRITE SYS$OUTPUT -                                       0506
$       " The directory or directories shown above should be checked for" 0507
$     WRITE SYS$OUTPUT -                                       0508
$       " files which are no longer required. Unnecessary files should" 0509
$     WRITE SYS$OUTPUT -                                       0510
$       " be deleted as soon as possible."                     0511
$   PROMPT:                                                    0512
$     ON WARNING THEN GOTO PROMPT                              0513
$     SET MESSAGE/FACILITY -                                    0514
$       /IDENTIFICATION -                                      0515
$       /SEVERITY -                                           0516
$       /TEXT                                                  0517
$     WRITE SYS$OUTPUT " "                                     0518
$     INQUIRE/NOPUNCTUATION RESPONSE -                        0519
$       " Please enter RETURN to continue ... "              0520
$     IF RESPONSE .EQS. "***" THEN GOTO PASS1                 0521

```

DCL Command Procedure LOGIN.COM (cont.)

\$	@DEV\$206:[BCDSSME2.FIPM]FIPM_MENU.COM	0522
\$	EXIT	0523
\$	PASS1:	0524
\$	WRITE SYS\$OUTPUT " "	0525
\$	SET TERMINAL/NOECHO	0526
\$	INQUIRE PASSWORD1 -	0527
\$	" Please enter PASSWORD 1"	0528
\$	SET TERMINAL/ECHO	0529
\$	IF PASSWORD1 .NES. "*****" THEN GOTO PROMPT	0530
\$	WRITE SYS\$OUTPUT " "	0531
\$	SET TERMINAL/NOECHO	0532
\$	INQUIRE PASSWORD2 -	0533
\$	" Please enter PASSWORD 2"	0534
\$	SET TERMINAL/ECHO	0535
\$	IF PASSWORD2 .NES. "*****" THEN GOTO PROMPT	0536
\$	EXIT	0537

APPENDIX E

FIPM DATATRIEVE COMMAND FILES

<u>Datatrieve Command File</u>	<u>Page</u>
LIST_CON_1.COM	E-3
LIST_CON_2.COM	E-4
LIST_FIP_1.COM	E-6
LIST_FIP_2.COM	E-7
LIST_FM_1.COM	E-9
LIST_FM_2.COM	E-10
LIST_MOD_1.COM	E-12
LIST_MOD_2.COM	E-13
LIST_REF_1.COM	E-15
LIST_REF_2.COM	E-16
LIST_SYS_1.COM	E-18
LIST_SYS_2.COM	E-19
MODIFY_FIP.COM	E-21
MODIFY_FM.COM	E-22
MODIFY_MOD.COM	E-23
MODIFY_REF.COM	E-24
MODIFY_SYS.COM	E-25
STORE_CON.COM	E-26
STORE_FIP.COM	E-27
STORE_FM.COM	E-28
STORE_MOD.COM	E-29
STORE_REF.COM	E-30
STORE_SYS.COM	E-31

E-2

(This page intentionally blank)

Datatrieve Command File LIST_CON_1.COM

DECLARE ICNT	PIC 9(5).	0001
DECLARE JCNT	PIC 9(5).	0002
DECLARE KCNT	PIC 9(5).	0003
READY CONNECTIONS SHARED READ		0004
ICNT = 0		0005
FOR CONNECTIONS ICNT = ICNT + 1		0006
JCNT = ICNT / 5		0007
KCNT = JCNT * 5		0008
IF KCNT LT ICNT THEN JCNT = JCNT + 1		0009
:CLRSCRN		0010
PRINT SKIP 3,		0011
"Domain CONNECTIONS contains", SPACE 1,		0012
ICNT (-) USING ZZZZ9, SPACE 1, "records.", SKIP 2,		0013
"With 5 records per page the resulting file", SKIP 1,		0014
"will contain", SPACE 1, JCNT (-) USING ZZZZ9, SPACE 1,		0015
"pages."		0016
EXIT		0017

Datatrieve Command File LIST_CON_2.COM

```

$ DTR32                                0018
SET NO PROMPT                          0019
!                                     0020
!                                     0021
!=====                             0022
!                                     0023
! PROCEDURE TO LIST THE RECORDS IN DOMAIN CONNECTIONS TO A FILE FOR 0024
! PRINTING. THE RECORDS ARE SORTED IN ASCENDING ORDER OF THE FOLLOWING 0025
! FIELDS:                             0026
!   1. SYSTEM_MODULE_A                0027
!   2. CONNECTION                     0028
!   3. UNANTICIPATED_CONNECTION       0029
!   4. SYSTEM_MODULE_B                0030
! THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN 80-COLUMN FORMAT 0031
! AND HAS NO MORE THAN 60 LINES PER PAGE. 0032
!=====                             0033
!                                     0034
!                                     0035
DECLARE ICNT          PIC 9(5).        0036
DECLARE JCNT          PIC 9(5).        0037
DECLARE KCNT          PIC 9(5).        0038
DECLARE CAL           USAGE DATE       0039
                        EDIT_STRING X(23). 0040
READY MODULES        SHARED READ      0041
READY CONNECTIONS SHARED READ          0042
JCNT = 0              0043
FOR CONNECTIONS JCNT = JCNT + 1        0044
CAL = "NOW"           0045
ON DEV$206:[BCDSSME2.LISTS]LIST_CON.LST 0046
BEGIN 0047
    ICNT = 1          0048
    KCNT = 1          0049
    FOR MODULES SORTED BY SYSTEM_MODULE 0050
    BEGIN 0051
        FOR CONNECTIONS WITH 0052
        SYSTEM_MODULE_A = MODULES.SYSTEM_MODULE SORTED BY 0053
        CODE_NUMBER 0054
        BEGIN 0055
            IF KCNT = 1 THEN PRINT NEW_PAGE, 0056
                                COL 1, "Domain CONNECTIONS", 0057
                                COL 64, CAL (-) USING X(17) 0058
            PRINT SKIP 2, 0059
            COL 1, "RECORD NO.", SPACE 1, 0060
                    ICNT (-) USING ZZZZ9, SPACE 1, 0061
                    "OF", SPACE 1, 0062
                    JCNT (-) USING ZZZZ9, SKIP 1, 0063
            COL 1, "=====", 0064
                    SPACE 0, 0065
                    "=====", 0066
                    SKIP 2, 0067
            COL 1, "DATE_CREATED", ":", SPACE 1, 0068

```

Datatrieve Command File LIST_CON_2.COM (cont.)

```

                                DATE_CREATED (-) USING X(23), SKIP 1,      0069
COL 1, "CODE_NUMBER      :",      SPACE 1,      0070
                                SYSTEM A (-) USING X(4),      0071
                                SPACE 1,      0072
                                MODULE A (-) USING 9(4),      0073
                                SPACE 1,      0074
                                CONNECTION_TYPE (-) USING X(2),      0075
                                SPACE 1,      0076
                                CONNECTION_QUALIFIER (-) USING X(2),      0077
                                SPACE 1,      0078
                                UNANTICIPATED_CONNECTION (-) USING X(1),      0079
                                SPACE 1,      0080
                                SYSTEM B (-) USING X(4),      0081
                                SPACE 1,      0082
                                MODULE_B (-) USING 9(4),      0083
                                SKIP 1,      0084
COL 1, "DATE_LAST_MODIFIED :",      SPACE 1,      0085
                                DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,      0086
COL 1, "MODIFYING_PROCEDURE :",      SPACE 1,      0087
                                MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,      0088
COL 1, "=====",      0089
                                SPACE 0,      0090
                                "=====",      0091
                                ICNT = ICNT + 1      0092
                                KCNT = KCNT + 1      0093
                                IF KCNT = 6 THEN KCNT = 1      0094
                                END      0095
                                END      0096
                                END      0097
                                EXIT      0098
                                $ EXIT      0099

```

Datatrieve Command File LIST_FIP_1.COM

:FIP_COUNT
EXIT

0100

0101

Datatrieve Command File LIST_FIP_2.COM

```

$ DTR32                                0102
SET NO PROMPT                          0103
!                                     0104
!                                     0105
=====                                0106
!                                     0107
!   PROCEDURE TO LIST THE RECORDS IN THE FAILURE INFORMATION PROPAGATIONS 0108
!   DOMAINS TO A FILE FOR PRINTING. THE RECORDS ARE SORTED IN ASCENDING 0109
!   ORDER OF THE FOLLOWING FIELDS FOR EACH DOMAIN:                        0110
!       1. FMCODE                                                         0111
!       2. SIGNAL_TYPE                                                    0112
!       3. PARAMETER                                                       0113
!       4. DATE_CREATED                                                    0114
!   THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN 80-COLUMN FORMAT 0115
!   AND HAS NO MORE THAN 60 LINES PER PAGE.                             0116
!                                     0117
=====                                0118
!                                     0119
DECLARE ICNT          PIC 9(5).      0120
DECLARE JCNT          PIC 9(5).      0121
DECLARE KCNT          PIC 9(5).      0122
DECLARE LCNT          PIC 9(5).      0123
DECLARE MCNT          PIC 9(5).      0124
DECLARE NCNT          PIC 9(5).      0125
DECLARE PCNT          PIC 9(5).      0126
DECLARE HSYSTEM       PIC X(4).      0127
DECLARE CAL           USAGE DATE     0128
                        EDIT_STRING X(23). 0129
READY SYSTEMS SHARED READ            0130
CAL = "NOW"                          0131
ICNT = 0                             0132
FOR SYSTEMS WITH PROPAGATIONS_FILE_CREATED = "YES" 0133
    ICNT = ICNT + 1                  0134
JCNT = 0                             0135
MCNT = 0                             0136
:FIP_LIST_1                          0137
JCNT = 0                             0138
NCNT = 0                             0139
:FIP_LIST_3                          0140
IF FN$TRANS_LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN 0141
    FN$DELETE_LOG("PROPAGATIONS") 0142
IF FN$TRANS_LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN 0143
    FN$DELETE_LOG("PROPAGATIONS_FILE") 0144
IF FN$TRANS_LOG("PROC") NE "PROC" THEN 0145
    FN$DELETE_LOG("PROC")          0146
EXIT                                0147
$ SET DEFAULT DEV$206:[BCDSSME2.LISTS] 0148
$ CREATE DEV$206:[BCDSSME2.LISTS]LIST_FIP.LST 0149
$ APPEND FIP *.LST LIST_FIP.LST      0150
$ SET PROTECTION=(OWNER:RWED) FIP_*.LST 0151
$ DELETE FIP_*.LST;*                 0152

```

Datatrieve Command File LIST_FIP_2.COM (cont.)

\$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWE,GROUP,WORLD) LIST_FIP.LST
\$ SET DEFAULT DEV\$206:[BCDSSME2.MISC]

0153

0154

Datatrieve Command File LIST_FM_1.COM

DECLARE ICNT	PIC 9(5).	0155
DECLARE JCNT	PIC 9(5).	0156
DECLARE KCNT	PIC 9(5).	0157
READY FAILUREMODES SHARED READ		0158
ICNT = 0		0159
FOR FAILUREMODES ICNT = ICNT + 1		0160
JCNT = ICNT / 2		0161
KCNT = JCNT * 2		0162
IF KCNT LT ICNT THEN JCNT = JCNT + 1		0163
:CLRSCRN		0164
PRINT SKIP 3,		0165
"Domain FAILUREMODES contains", SPACE 1,		0166
ICNT (-) USING ZZZZ9, SPACE 1, "records.", SKIP 2,		0167
"With 2 records per page the resulting file", SKIP 1,		0168
"will contain", SPACE 1, JCNT (-) USING ZZZZ9, SPACE 1,		0169
"pages."		0170
EXIT		0171

Datatrieve Command File LIST_FM_2.COM

```

$ DTR32                                0172
SET NO PROMPT                          0173
!                                     0174
!                                     0175
!=====                             0176
!                                     0177
! PROCEDURE TO LIST THE RECORDS IN DOMAIN FAILUREMODES TO A FILE FOR 0178
! PRINTING. THE RECORDS ARE SORTED IN ASCENDING ORDER OF THE FIELD 0179
! FMCODE. THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN 80-COLUMN 0180
! FORMAT AND HAS NO MORE THAN 60 LINES PER PAGE. 0181
!=====                             0182
!                                     0183
!                                     0184
DECLARE ICNT          PIC 9(5).        0185
DECLARE JCNT          PIC 9(5).        0186
DECLARE KCNT          PIC 9(5).        0187
DECLARE CAL           USAGE DATE       0188
                                EDIT STRING X(23). 0189
READY FAILUREMODES SHARED READ        0190
JCNT = 0                              0191
FOR FAILUREMODES JCNT = JCNT + 1      0192
CAL = "NOW"                           0193
ON DEV$206:[BCDSSME2.LISTS]LIST_FM.LST 0194
  BEGIN                               0195
    ICNT = 1                          0196
    KCNT = 1                          0197
    FOR FAILUREMODES SORTED BY FMCODE 0198
      BEGIN                           0199
        IF KCNT = 1 THEN PRINT NEW_PAGE, 0200
                                COL 1, "Domain FAILUREMODES", 0201
                                COL 64, CAL (-) USING X(17) 0202
        PRINT SKIP 1,               0203
          COL 1, "RECORD NO.",      SPACE 1, 0204
          ICNT (-) USING ZZZZ9, SPACE 1, 0205
          "OF",                     SPACE 1, 0206
          JCNT (-) USING ZZZZ9, SKIP 1, 0207
          COL 1, "=====", 0208
          SPACE 0, 0209
          "=====", 0210
          SKIP 2, 0211
          COL 1, "DATE_CREATED      :", SPACE 1, 0212
          DATE_CREATED (-) USING X(23), SKIP 1, 0213
          COL 1, "FMCODE           :", SPACE 1, 0214
          SOURCE_SYSTEM (-) USING X(4), SPACE 1, 0215
          SOURCE_MODULE (-) USING 9(4), SPACE 1, 0216
          FAILURE_MODE (-) USING X(2), SPACE 1, 0217
          FAILURE_SUBMODE (-) USING X(2), SPACE 1, 0218
          ACCOMPLICE_SYSTEM (-) USING X(4), SPACE 1, 0219
          ACCOMPLICE_MODULE (-) USING 9(4), SKIP 1, 0220
          COL 1, "DESCRIPTION      :", SPACE 1, 0221
          DESCRIPTION (-) USING T(58), SKIP 1, 0222

```

Datatrieve Command File LIST_FM_2.COM (cont.)

```

COL 1, "EFFECT1          :", SPACE 1,          0223
      EFFECT1 (-) USING T(58), SKIP 1,          0224
COL 1, "EFFECT2          :", SPACE 1,          0225
      EFFECT2 (-) USING T(58), SKIP 1,          0226
COL 1, "EFFECT3          :", SPACE 1,          0227
      EFFECT3 (-) USING T(58), SKIP 1,          0228
COL 1, "EFFECT4          :", SPACE 1,          0229
      EFFECT4 (-) USING T(58), SKIP 1,          0230
COL 1, "EFFECT5          :", SPACE 1,          0231
      EFFECT5 (-) USING T(58), SKIP 1,          0232
COL 1, "EFFECT6          :", SPACE 1,          0233
      EFFECT6 (-) USING T(58), SKIP 1,          0234
COL 1, "DATE_LAST_MODIFIED :", SPACE 1,          0235
      DATE_LAST_MODIFIED (-) USING X(23), SKIP 1, 0236
COL 1, "MODIFYING_PROCEDURE :", SPACE 1,          0237
      MODIFYING_PROCEDURE (-) USING X(20), SKIP 2, 0238
COL 1, "===== ",          0239
      SPACE 0,          0240
      "===== "          0241
      ICNT = ICNT + 1          0242
      KCNT = KCNT + 1          0243
      IF KCNT = 3 THEN KCNT = 1 0244
END          0245
          0246
END          0247
EXIT          0248
$ EXIT

```

Datatrieve Command File LIST_MOD_1.COM

DECLARE ICNT	PIC 9(5).	0249
DECLARE JCNT	PIC 9(5).	0250
DECLARE KCNT	PIC 9(5).	0251
READY MODULES SHARED READ		0252
ICNT = 0		0253
FOR MODULES ICNT = ICNT + 1		0254
JCNT = ICNT / 3		0255
KCNT = JCNT * 3		0256
IF KCNT LT ICNT THEN JCNT = JCNT + 1		0257
:CLRSCRN		0258
PRINT SKIP 3,		0259
"Domain MODULES contains", SPACE 1,		0260
ICNT (-) USING ZZZZ9, SPACE 1, "records.", SKIP 2,		0261
"With 3 records per page the resulting file", SKIP 1,		0262
"will contain", SPACE 1, JCNT (-) USING ZZZZ9, SPACE 1,		0263
"pages."		0264
EXIT		0265

Datatrieve Command File LIST_MOD_2.COM

```

$ DTR32
SET NO PROMPT
!
!
! =====
! PROCEDURE TO LIST THE RECORDS IN DOMAIN MODULES TO A FILE FOR
! PRINTING. THE RECORDS ARE SORTED IN ASCENDING ORDER OF THE FIELD
! SYSTEM MODULE. THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN
! 80-COLUMN FORMAT AND HAS NO MORE THAN 60 LINES PER PAGE.
! =====
!
DECLARE ICNT PIC 9(5).
DECLARE JCNT PIC 9(5).
DECLARE KCNT PIC 9(5).
DECLARE CAL USAGE DATE
EDIT_STRING X(23).
READY MODULES SHARED READ
JCNT = 0
FOR MODULES JCNT = JCNT + 1
CAL = "NOW"
ON DEV$206:[BCDSSME2.LISTS]LIST_MOD.LST
BEGIN
    ICNT = 1
    KCNT = 1
    FOR MODULES SORTED BY SYSTEM_MODULE
    BEGIN
        IF KCNT = 1 THEN PRINT NEW_PAGE,
            COL 1, "Domain MODULES",
            COL 64, CAL (-) USING X(17)
        PRINT SKIP 2,
            COL 1, "RECORD NO.", SPACE 1,
            ICNT (-) USING ZZZZ9, SPACE 1,
            "OF", SPACE 1,
            JCNT (-) USING ZZZZ9, SKIP 1,
            COL 1, "===== ",
            SPACE 0,
            "===== ",
            SKIP 2,
            COL 1, "DATE_CREATED :", SPACE 1,
            DATE_CREATED (-) USING X(23), SKIP 1,
            COL 1, "SYSTEM_MODULE :", SPACE 1,
            SYSTEM (-) USING X(4), SPACE 1,
            MODULE (-) USING 9(4), SKIP 1,
            COL 1, "SYSTEM_MODULE_NAME :", SPACE 1,
            SYSTEM_MODULE_NAME (-) USING T(55), SKIP 1,
            COL 1, "SYSTEM_MODULE_FUNCTION :", SPACE 1,
            SYSTEM_MODULE_FUNCTION (-) USING T(55), SKIP 1,
            COL 1, "DATE_LAST_MODIFIED :", SPACE 1,
            DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,

```

Datatrieve Command File LIST_MOD_2.COM (cont.)

COL 1, "MODIFYING_PROCEDURE :",	SPACE 1,	0317
MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,		0318
COL 1, "=====		0319
SPACE 0,		0320
"=====		0321
ICNT = ICNT + 1		0322
KCNT = KCNT + 1		0323
IF KCNT = 4 THEN KCNT = 1		0324
END		0325
END		0326
EXIT		0327
\$ EXIT		0328

Datatrieve Command File LIST_REF_1.COM

DECLARE ICNT	PIC 9(5).	0329
DECLARE JCNT	PIC 9(5).	0330
DECLARE KCNT	PIC 9(5).	0331
READY REFERENCES	SHARED READ	0332
ICNT = 0		0333
FOR REFERENCES	ICNT = ICNT + 1	0334
JCNT = ICNT / 2		0335
KCNT = JCNT * 2		0336
IF KCNT LT ICNT	THEN JCNT = JCNT + 1	0337
:CLRSCRN		0338
PRINT SKIP 3,		0339
"Domain REFERENCES contains",	SPACE 1,	0340
ICNT (-) USING ZZZZ9,	SPACE 1, "records.",	0341
SKIP 2,		
"With 2 records per page	the resulting file",	0342
SKIP 1,		
"will contain",	SPACE 1, JCNT (-) USING	0343
ZZZZ9,	SPACE 1,	
"pages."		0344
EXIT		0345

Datatrieve Command File LIST_REF_2.COM

```

$ DTR32
SET NO PROMPT
!
!
=====
!
! PROCEDURE TO LIST THE RECORDS IN DOMAIN REFERENCES TO A FILE FOR
! PRINTING. THE RECORDS ARE SORTED IN ASCENDING ORDER OF THE FIELD
! REFERENCE_NUMBER. THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN
! 80-COLUMN FORMAT AND HAS NO MORE THAN 60 LINES PER PAGE.
!
=====
!
DECLARE ICNT          PIC 9(5).
DECLARE JCNT          PIC 9(5).
DECLARE KCNT          PIC 9(5).
DECLARE CAL           USAGE DATE
                     EDIT STRING X(23).
READY REFERENCES SHARED READ
JCNT = 0
FOR REFERENCES JCNT = JCNT + 1
CAL = "NOW"
ON DEV$206:[BCDSSME2.LISTS]LIST_REF.LST
BEGIN
    ICNT = 1
    KCNT = 1
    FOR REFERENCES SORTED BY REFERENCE_NUMBER
    BEGIN
        IF KCNT = 1 THEN PRINT NEW_PAGE,
                                COL 1, "Domain REFERENCES",
                                COL 64, CAL (-) USING X(17)
        PRINT SKIP 4,
            COL 1, "RECORD NO.",          SPACE 1,
                    ICNT (-) USING ZZZZ9, SPACE 1,
                    "OF",                  SPACE 1,
                    JCNT (-) USING ZZZZ9, SKIP 1,
            COL 1, "===== ",
                    SPACE 0,
                    "===== ",
                    SKIP 2,
            COL 1, "DATE_CREATED      :",          SPACE 1,
                    DATE_CREATED (-) USING X(23), SKIP 1,
            COL 1, "REFERENCE_NUMBER  :",          SPACE 1,
                    REFERENCE_NUMBER (-) USING X(5), SKIP 1,
            COL 1, "AUTHOR1           :",          SPACE 1,
                    AUTHOR1 (-) USING X(25), SKIP 1,
            COL 1, "AUTHOR2           :",          SPACE 1,
                    AUTHOR2 (-) USING X(25), SKIP 1,
            COL 1, "AUTHOR3           :",          SPACE 1,
                    AUTHOR3 (-) USING X(25), SKIP 1,
            COL 1, "AUTHOR4           :",          SPACE 1,

```


Datatrieve Command File LIST_REF_2.COM (cont.)

	AUTHOR4 (-) USING X(25), SKIP 1,	0397
COL 1,	"DOCUMENT_TITLE :", SPACE 1,	0398
	DOCUMENT_TITLE (-) USING T(58), SKIP 1,	0399
COL 1,	"DOCUMENT_SOURCE :", SPACE 1,	0400
	DOCUMENT_SOURCE (-) USING X(30), SKIP 1,	0401
COL 1,	"DOCUMENT_NUMBER :", SPACE 1,	0402
	DOCUMENT_NUMBER (-) USING X(30), SKIP 1,	0403
COL 1,	"DOCUMENT_DATE :", SPACE 1,	0404
	DOCUMENT_DATE (-) USING X(11), SKIP 1,	0405
COL 1,	"CONTRACT_NUMBER :", SPACE 1,	0406
	CONTRACT_NUMBER (-) USING X(20), SKIP 1,	0407
COL 1,	"DATE_LAST_MODIFIED :", SPACE 1,	0408
	DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,	0409
COL 1,	"MODIFYING_PROCEDURE :", SPACE 1,	0410
	MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,	0411
COL 1,	"=====",	0412
	SPACE 0,	0413
	"=====	0414
	ICNT = ICNT + 1	0415
	KCNT = KCNT + 1	0416
	IF KCNT = 3 THEN KCNT = 1	0417
	END	0418
END		0419
EXIT		0420
\$ EXIT		0421

Datatrieve Command File LIST_SYS_1.COM

DECLARE ICNT	PIC 9(5).	0422
DECLARE JCNT	PIC 9(5).	0423
DECLARE KCNT	PIC 9(5).	0424
READY SYSTEMS SHARED READ		0425
ICNT = 0		0426
FOR SYSTEMS ICNT = ICNT + 1		0427
JCNT = ICNT / 2		0428
KCNT = JCNT * 2		0429
IF KCNT LT ICNT THEN JCNT = JCNT + 1		0430
:CLRSCRN		0431
PRINT SKIP 3,		0432
"Domain SYSTEMS contains", SPACE 1,		0433
ICNT (-) USING ZZZZ9, SPACE 1, "records.", SKIP 2,		0434
"With 2 records per page the resulting file", SKIP 1,		0435
"will contain", SPACE 1, JCNT (-) USING ZZZZ9, SPACE 1,		0436
"pages."		0437
EXIT		0438

Datatrieve Command File LIST_SYS_2.COM

```

$ DTR32
SET NO PROMPT
!
!
!-----
!
! PROCEDURE TO LIST THE RECORDS IN DOMAIN SYSTEMS TO A FILE FOR
! PRINTING. THE RECORDS ARE SORTED IN ASCENDING ORDER OF THE FIELD
! SYSTEM. THE RESULTING FILE IS DESIGNED TO BE PRINTED IN AN 80-COLUMN
! FORMAT AND HAS NO MORE THAN 60 LINES PER PAGE.
!-----
!
DECLARE ICNT          PIC 9(5).
DECLARE JCNT          PIC 9(5).
DECLARE KCNT          PIC 9(5).
DECLARE CAL           USAGE DATE
                     EDIT_STRING X(23).

READY SYSTEMS SHARED READ
JCNT = 0
FOR SYSTEMS JCNT = JCNT + 1
CAL = "NOW"
ON DEV$206:[BCDSSME2.LISTS]LIST_SYS.LST
  BEGIN
    ICNT = 1
    KCNT = 1
    FOR SYSTEMS SORTED BY SYSTEM
      BEGIN
        IF KCNT = 1 THEN PRINT NEW_PAGE,
                           COL 1, "Domain SYSTEMS",
                           COL 64, CAL (-) USING X(17)
        PRINT SKIP 4,
        COL 1, "RECORD NO.",          SPACE 1,
              ICNT (-) USING ZZZZ9, SPACE 1,
              "OF",                   SPACE 1,
              JCNT (-) USING ZZZZ9, SKIP 1,
        COL 1, "===== ",
              SPACE 0,
              "===== ",
              SKIP 2,
        COL 1, "DATE_CREATED          :",          SPACE 1,
              DATE_CREATED (-) USING X(23), SKIP 1,
        COL 1, "SYSTEM                :",          SPACE 1,
              SYSTEM (-) USING X(4), SKIP 1,
        COL 1, "SYSTEM_NAME           :",          SPACE 1,
              SYSTEM_NAME (-) USING T(58), SKIP 1,
        COL 1, "FMEA ITEMS             :",
              COL 23, " 1)", SPACE 1, ITEM1 (-) USING X(4),
              COL 42, " 6)", SPACE 1, ITEM6 (-) USING X(4),
              COL 61, "11)", SPACE 1, ITEM11 (-) USING X(4),
              SKIP 1,

```

Datatrieve Command File LIST_SYS_2.COM (cont.)

```

COL 23, " 2)", SPACE 1, ITEM2 (-) USING X(4), 0490
COL 42, " 7)", SPACE 1, ITEM7 (-) USING X(4), 0491
COL 61, "12)", SPACE 1, ITEM12 (-) USING X(4), 0492
SKIP 1, 0493
COL 23, " 3)", SPACE 1, ITEM3 (-) USING X(4), 0494
COL 42, " 8)", SPACE 1, ITEM8 (-) USING X(4), 0495
COL 61, "13)", SPACE 1, ITEM13 (-) USING X(4), 0496
SKIP 1, 0497
COL 23, " 4)", SPACE 1, ITEM4 (-) USING X(4), 0498
COL 42, " 9)", SPACE 1, ITEM9 (-) USING X(4), 0499
COL 61, "14)", SPACE 1, ITEM14 (-) USING X(4), 0500
SKIP 1, 0501
COL 23, " 5)", SPACE 1, ITEM5 (-) USING X(4), 0502
COL 42, "10)", SPACE 1, ITEM10 (-) USING X(4), 0503
COL 61, "15)", SPACE 1, ITEM15 (-) USING X(4), 0504
SKIP 1, 0505
COL 1, "REFERENCES :", 0506
COL 23, " 1)", SPACE 1, REFERENCE1 (-) USING X(5), 0507
COL 42, " 5)", SPACE 1, REFERENCE5 (-) USING X(5), 0508
COL 61, " 9)", SPACE 1, REFERENCE9 (-) USING X(5), 0509
SKIP 1, 0510
COL 23, " 2)", SPACE 1, REFERENCE2 (-) USING X(5), 0511
COL 42, " 6)", SPACE 1, REFERENCE6 (-) USING X(5), 0512
COL 61, "10)", SPACE 1, REFERENCE10 (-) USING X(5), 0513
SKIP 1, 0514
COL 23, " 3)", SPACE 1, REFERENCE3 (-) USING X(5), 0515
COL 42, " 7)", SPACE 1, REFERENCE7 (-) USING X(5), 0516
SKIP 1, 0517
COL 23, " 4)", SPACE 1, REFERENCE4 (-) USING X(5), 0518
COL 42, " 8)", SPACE 1, REFERENCE8 (-) USING X(5), 0519
SKIP 1, 0520
COL 1, "PROPAGATIONS_FILE_", SKIP 1, 0521
COL 1, "   CREATED :", SPACE 1, 0522
PROPAGATIONS_FILE_CREATED (-) USING X(3), SKIP 1, 0523
COL 1, "DATE_LAST_MODIFIED :", SPACE 1, 0524
DATE_LAST_MODIFIED (-) USING X(23), SKIP 1, 0525
COL 1, "MODIFYING_PROCEDURE :", SPACE 1, 0526
MODIFYING_PROCEDURE (-) USING X(20), SKIP 2, 0527
COL 1, "===== ", 0528
SPACE 0, 0529
"===== " 0530
ICNT = ICNT + 1 0531
KCNT = KCNT + 1 0532
IF KCNT = 3 THEN KCNT = 1 0533
END 0534
END 0535
EXIT 0536
$ EXIT 0537

```

Datatrieve Command File MODIFY_FIP.COM

OPEN DEV\$206:[BCDSSME2.LOGS]MODIFY_FIP.LST	0538
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0539
CAL = "NOW"	0540
PRINT NEW_PAGE	0541
:CLRSCRN	0542
PRINT SKIP 2,	0543
" MODIFY PROPAGATIONS", SKIP 2,	0544
"=====", SKIP 2,	0545
"===== START: ", SPACE 0,	0546
CAL (-) USING X(17),	0547
SPACE 0, " =====", SKIP 2,	0548
"=====", SKIP 3	0549
:FIP MODIFY	0550
CAL = "NOW"	0551
PRINT NEW_PAGE	0552
:CLRSCRN	0553
PRINT SKIP 2,	0554
" MODIFY PROPAGATIONS", SKIP 2,	0555
"=====", SKIP 2,	0556
"===== END: ", SPACE 0,	0557
CAL (-) USING X(17),	0558
SPACE 0, " =====", SKIP 2,	0559
"=====", SKIP 3	0560
CLOSE	0561
RELEASE	0562
FINISH	0563
EXIT	0564

Datatrieve Command File MODIFY_FM.COM

OPEN DEV\$206:[BCDSSME2.LOGS]MODIFY_FM.LST	0565
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0566
CAL = "NOW"	0567
PRINT NEW_PAGE	0568
:CLRSCRN	0569
PRINT SKIP 2,	0570
" MODIFY FAILURE MODES", SKIP 2,	0571
"=====", SKIP 2,	0572
"===== START: ", SPACE 0,	0573
CAL (-) USING X(17),	0574
SPACE 0, " =====", SKIP 2,	0575
"=====", SKIP 3	0576
:FM_MODIFY	0577
CAL = "NOW"	0578
PRINT NEW_PAGE	0579
:CLRSCRN	0580
PRINT SKIP 2,	0581
" MODIFY FAILURE MODES", SKIP 2,	0582
"=====", SKIP 2,	0583
"===== END: ", SPACE 0,	0584
CAL (-) USING X(17),	0585
SPACE 0, " =====", SKIP 2,	0586
"=====", SKIP 3	0587
CLOSE	0588
RELEASE	0589
FINISH	0590
EXIT	0591

Datatrieve Command File MODIFY_MOD.COM

OPEN DEV\$206:[BCDSSME2.LOGS]MODIFY_MOD.LST	0592
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0593
CAL = "NOW"	0594
PRINT NEW_PAGE	0595
:CLRSCRN	0596
PRINT SKIP 2,	0597
" MODIFY MODULES", SKIP 2,	0598
"=====", SKIP 2,	0599
"===== START: ", SPACE 0,	0600
CAL (-) USING X(17),	0601
SPACE 0, " =====", SKIP 2,	0602
"=====", SKIP 3	0603
:MOD_MODIFY	0604
CAL = "NOW"	0605
PRINT NEW_PAGE	0606
:CLRSCRN	0607
PRINT SKIP 2,	0608
" MODIFY MODULES", SKIP 2,	0609
"=====", SKIP 2,	0610
"===== END: ", SPACE 0,	0611
CAL (-) USING X(17),	0612
SPACE 0, " =====", SKIP 2,	0613
"=====", SKIP 3	0614
CLOSE	0615
RELEASE	0616
FINISH	0617
EXIT	0618

Datatrieve Command File MODIFY_REF.COM

OPEN DEV\$206:[BCDSSME2.LOGS]MODIFY_REF.LST	0619
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0620
CAL = "NOW"	0621
PRINT NEW_PAGE	0622
:CLRSCRN	0623
PRINT SKIP 2;	0624
" MODIFY REFERENCES", SKIP 2,	0625
"===== ", SKIP 2,	0626
"===== START: ", SPACE 0,	0627
CAL (-) USING X(17),	0628
SPACE 0, " =====", SKIP 2,	0629
"===== ", SKIP 3	0630
:REF MODIFY	0631
CAL = "NOW"	0632
PRINT NEW_PAGE	0633
:CLRSCRN	0634
PRINT SKIP 2,	0635
" MODIFY REFERENCES", SKIP 2,	0636
"===== ", SKIP 2,	0637
"===== END: ", SPACE 0,	0638
CAL (-) USING X(17),	0639
SPACE 0, " =====", SKIP 2,	0640
"===== ", SKIP 3	0641
CLOSE	0642
RELEASE	0643
FINISH	0644
EXIT	0645

Datatrieve Command File MODIFY_SYS.COM

OPEN DEV\$206:[BCDSSME2.LOGS]MODIFY_SYS.LST	0646
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0647
CAL = "NOW"	0648
PRINT NEW_PAGE	0649
:CLRSCRN	0650
PRINT SKIP 2,	0651
" MODIFY SYSTEMS", SKIP 2,	0652
"=====", SKIP 2,	0653
"==== START: ", SPACE 0,	0654
CAL (-) USING X(17),	0655
SPACE 0, " =====", SKIP 2,	0656
"=====", SKIP 3	0657
:SYS MODIFY	0658
CAL = "NOW"	0659
PRINT NEW_PAGE	0660
:CLRSCRN	0661
PRINT SKIP 2,	0662
" MODIFY SYSTEMS", SKIP 2,	0663
"=====", SKIP 2,	0664
"==== END: ", SPACE 0,	0665
CAL (-) USING X(17),	0666
SPACE 0, " =====", SKIP 2,	0667
"=====", SKIP 3	0668
CLOSE	0669
RELEASE	0670
FINISH	0671
EXIT	0672

Datatrieve Command File STORE_CON.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE CON.LST	0673
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0674
CAL = "NOW"	0675
PRINT NEW_PAGE	0676
:CLRSCRN	0677
PRINT SKIP 2,	0678
" STORE CONNECTIONS", SKIP 2,	0679
"=====", SKIP 2,	0680
"===== START: ", SPACE 0,	0681
CAL (-) USING X(17),	0682
SPACE 0, " =====", SKIP 2,	0683
"=====", SKIP 3	0684
:CON_STORE	0685
CAL = "NOW"	0686
PRINT NEW_PAGE	0687
:CLRSCRN	0688
PRINT SKIP 2,	0689
" STORE CONNECTIONS", SKIP 2,	0690
"=====", SKIP 2,	0691
"===== END: ", SPACE 0,	0692
CAL (-) USING X(17),	0693
SPACE 0, " =====", SKIP 2,	0694
"=====", SKIP 3	0695
CLOSE	0696
RELEASE	0697
FINISH	0698
EXIT	0699

Datatrieve Command File STORE_FIP.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE_FIP.LST	0700
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0701
CAL = "NOW"	0702
PRINT NEW_PAGE	0703
:CLRSCRN	0704
PRINT SKIP 2,	0705
" STORE PROPAGATIONS", SKIP 2,	0706
"=====", SKIP 2,	0707
"===== START: ", SPACE 0,	0708
CAL (-) USING X(17),	0709
SPACE 0, " =====", SKIP 2,	0710
"=====", SKIP 3	0711
:FIP_STORE	0712
CAL = "NOW"	0713
PRINT NEW_PAGE	0714
:CLRSCRN	0715
PRINT SKIP 2,	0716
" STORE PROPAGATIONS", SKIP 2,	0717
"=====", SKIP 2,	0718
"===== END: ", SPACE 0,	0719
CAL (-) USING X(17),	0720
SPACE 0, " =====", SKIP 2,	0721
"=====", SKIP 3	0722
CLOSE	0723
RELEASE	0724
FINISH	0725
EXIT	0726

Datatrieve Command File STORE_FM.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE FM.LST	0727
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0728
CAL = "NOW"	0729
PRINT NEW_PAGE	0730
:CLRSCRN	0731
PRINT SKIP 2,	0732
" STORE FAILURE MODES", SKIP 2,	0733
"=====", SKIP 2,	0734
"===== START: ", SPACE 0,	0735
CAL (-) USING X(17),	0736
SPACE 0, " =====", SKIP 2,	0737
"=====", SKIP 3	0738
:FM_STORE	0739
CAL = "NOW"	0740
PRINT NEW_PAGE	0741
:CLRSCRN	0742
PRINT SKIP 2,	0743
" STORE FAILURE MODES", SKIP 2,	0744
"=====", SKIP 2,	0745
"===== END: ", SPACE 0,	0746
CAL (-) USING X(17),	0747
SPACE 0, " =====", SKIP 2,	0748
"=====", SKIP 3	0749
CLOSE	0750
RELEASE	0751
FINISH	0752
EXIT	0753

Datatrieve Command File STORE_MOD.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE_MOD.LST	0754
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0755
CAL = "NOW"	0756
PRINT NEW_PAGE	0757
:CLRSCRN	0758
PRINT SKIP 2,	0759
" STORE MODULES", SKIP 2,	0760
"=====", SKIP 2,	0761
"==== START: ", SPACE 0,	0762
CAL (-) USING X(17),	0763
SPACE 0, " =====", SKIP 2,	0764
"=====", SKIP 3	0765
:MOD_STORE	0766
CAL = "NOW"	0767
PRINT NEW_PAGE	0768
:CLRSCRN	0769
PRINT SKIP 2,	0770
" STORE MODULES", SKIP 2,	0771
"=====", SKIP 2,	0772
"==== END: ", SPACE 0,	0773
CAL (-) USING X(17),	0774
SPACE 0, " =====", SKIP 2,	0775
"=====", SKIP 3	0776
CLOSE	0777
RELEASE	0778
FINISH	0779
EXIT	0780

Datatrieve Command File STORE_REF.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE_REF.LST	0781
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0782
CAL = "NOW"	0783
PRINT NEW_PAGE	0784
:CLRSCRN	0785
PRINT SKIP 2,	0786
" STORE REFERENCES", SKIP 2,	0787
"=====", SKIP 2,	0788
"===== START: ", SPACE 0,	0789
CAL (-) USING X(17),	0790
SPACE 0, " =====", SKIP 2,	0791
"=====", SKIP 3	0792
:REF_STORE	0793
CAL = "NOW"	0794
PRINT NEW_PAGE	0795
:CLRSCRN	0796
PRINT SKIP 2,	0797
" STORE REFERENCES", SKIP 2,	0798
"=====", SKIP 2,	0799
"===== END: ", SPACE 0,	0800
CAL (-) USING X(17),	0801
SPACE 0, " =====", SKIP 2,	0802
"=====", SKIP 3	0803
CLOSE	0804
RELEASE	0805
FINISH	0806
EXIT	0807

Datatrieve Command File STORE_SYS.COM

OPEN DEV\$206:[BCDSSME2.LOGS]STORE_SYS.LST	0808
DECLARE CAL USAGE DATE EDIT_STRING X(23).	0809
CAL = "NOW"	0810
PRINT NEW_PAGE	0811
:CLRSCRN	0812
PRINT SKIP 2,	0813
" STORE SYSTEMS", SKIP 2,	0814
"=====", SKIP 2,	0815
"===== START: ", SPACE 0,	0816
CAL (-) USING X(17),	0817
SPACE 0, " =====", SKIP 2,	0818
"=====", SKIP 3	0819
:SYS_STORE	0820
CAL = "NOW"	0821
PRINT NEW_PAGE	0822
:CLRSCRN	0823
PRINT SKIP 2,	0824
" STORE SYSTEMS", SKIP 2,	0825
"=====", SKIP 2,	0826
"===== END: ", SPACE 0,	0827
CAL (-) USING X(17),	0828
SPACE 0, " =====", SKIP 2,	0829
"=====", SKIP 3	0830
CLOSE	0831
RELEASE	0832
FINISH	0833
EXIT	0834

E-32

(This page intentionally blank)

APPENDIX F

FIPM DATATRIEVE PROCEDURES

NOTE

The character strings "<ESC>" and "↑G" appear in some of the following procedures. These strings represent the ASCII escape and bell characters respectively. They are actually single characters which can be observed only while editing the appropriate file using the VAX EDT editor. To insert these characters in a file, the EDT special insert (SPECINS) function must be used.

<u>Datatrieve Procedure</u>	<u>Page</u>
BELL	F-3
CLRSCRN	F-4
CON_STORE	F-5
CREATE_CONNECTIONS	F-15
CREATE_CONNECTIONS_FORM	F-16
CREATE_FAILUREMODES	F-17
CREATE_FAILUREMODES_FORM	F-18
CREATE_MODULES	F-19
CREATE_MODULES_FORM	F-20
CREATE_PROPAGATIONS	F-21
CREATE_PROPAGATIONS_FIP_1	F-22
CREATE_PROPAGATIONS_FIP_2	F-23
CREATE_PROPAGATIONS_FORM	F-24
CREATE_PROPAGATIONS_SYS_1	F-25
CREATE_PROPAGATIONS_SYS_2	F-26
CREATE_REFERENCES	F-27
CREATE_REFERENCES_FORM	F-28
CREATE_SYSTEMS	F-29
CREATE_SYSTEMS_FORM	F-30
DTR_NULL	F-31
FIPLOGICALC	F-32

FIPLOGICALD	F-33
FIP_COUNT	F-34
FIP_COUNT_1	F-35
FIP_COUNT_2	F-36
FIP_LIST_1	F-37
FIP_LIST_2	F-38
FIP_LIST_3	F-39
FIP_LIST_4	F-40
FIP_MODIFY	F-42
FIP_MODIFY_1	F-48
FIP_MODIFY_2	F-52
FIP_MODIFY_3	F-70
FIP_MODIFY_4	F-72
FIP_STORE	F-82
FIP_STORE_1	F-87
FIP_STORE_2	F-89
FM_MODIFY	F-101
FM_MODIFY_1	F-120
FM_STORE	F-133
HDR	F-149
MOD_MODIFY	F-150
MOD_MODIFY_1	F-162
MOD_STORE	F-171
PRNTOFF	F-180
PRNTON	F-181
REF_MODIFY	F-182
REF_MODIFY_1	F-196
REF_STORE	F-205
S132	F-217
S80	F-218
SYS_MODIFY	F-219
SYS_MODIFY_1	F-231
SYS_MODIFY_2	F-233
SYS_STORE	F-247

Datatrieve Procedure BELL

DEFINE PROCEDURE BELL	0001
=====	0002
!	0003
! THIS PROCEDURE SOUNDS THE TERMINAL BELL BY PRINTING A SPECIAL CHARACTER	0004
!	0005
! THE CHARACTER USED IS "↑G" WHICH CORRESPONDS TO [CTRL/G]. THIS CHARACTER	0006
! MUST BE INSERTED IN THE PROCEDURE USING THE EDT OR TPU EDITORS SPECINS	0007
! FEATURE.	0008
!	0009
=====	0010
ON TT:	0011
PRINT "↑G"	0012
END-PROCEDURE	0013

Datatrieve Procedure CLRSCRN

```
DEFINE PROCEDURE CLRSCRN                                0014
!                                                         0015
!                                                         0016
!===== 0017
!                                                         0018
! PROCEDURE TO CLEAR THE TERMINAL SCREEN AND POSITION THE CURSOR IN THE 0019
! UPPER LEFT HAND CORNER. TO ACCOMPLISH THIS OBJECTIVE, TWO ANSI ESCAPE 0020
! SEQUENCES ARE PRINTED TO THE TERMINAL. THE ESCAPE SEQUENCES ARE AS 0021
! FOLLOWS: 0022
!                                                         0023
! 1. <ESC> [ 2 J      (CLEARS THE SCREEN) 0024
!                                                         0025
! 2. <ESC> [ 1 ; 1 H (POSITIONS THE CURSOR AT LINE 1; COLUMN 1) 0026
!                                                         0027
!===== 0028
!                                                         0029
ON TT: 0030
  PRINT "<ESC>[2J", "<ESC>[1;1H" 0031
END-PROCEDURE 0032
```

Datatrieve Procedure CON_STORE

```

DEFINE PROCEDURE CON_STORE                                0033
!                                                         0034
!                                                         0035
!=====                                                 0036
!                                                         0037
!  VARIABLES ASSOCIATED WITH THE SIX INPUT FIELDS FOR    0038
!  DOMAIN CONNECTIONS                                    0039
!    1. SYSTEM_A                                         0040
!    2. MODULE_A                                         0041
!    3. CONNECTION                                         0042
!    4. UNANTICIPATED_CONNECTION                         0043
!    5. SYSTEM_B                                         0044
!    6. MODULE_B                                         0045
!=====                                                 0046
!                                                         0047
!                                                         0048
DECLARE TSYSTEMA      PIC X(4).                           0049
DECLARE TMODULEA      PIC 9(4).                           0050
DECLARE TTMODULEA     PIC X(4).                           0051
DECLARE TCONNECT      PIC X(4).                           0052
DECLARE TUCONNECT     PIC X(1).                           0053
DECLARE TSYSTEMB      PIC X(4).                           0054
DECLARE TMODULEB      PIC 9(4).                           0055
DECLARE TTMODULEB     PIC X(4).                           0056
DECLARE TSYSTEM       PIC X(4).                           0057
DECLARE TMODULE       PIC 9(4).                           0058
DECLARE TTMODULE      PIC X(4).                           0059
!                                                         0060
!                                                         0061
!=====                                                 0062
!                                                         0063
!  VARIABLES USED AS FLAGS OR CONDITION INDICATORS       0064
!=====                                                 0065
!                                                         0066
!                                                         0067
DECLARE TCONTINUE     PIC X(1).                           0068
DECLARE IMSG          PIC 9(2).                           0069
DECLARE FLG1          PIC X(1).                           0070
DECLARE FLG2          PIC X(1).                           0071
!                                                         0072
!                                                         0073
!=====                                                 0074
!                                                         0075
!  VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE  0076
!  TDMS FORM                                             0077
!=====                                                 0078
!                                                         0079
DECLARE TMSG          PIC X(80).                          0080
DECLARE TMSG1         PIC X(80).                          0081
DECLARE TMSG2         PIC X(80).                          0082
DECLARE TMSG3         PIC X(80).                          0083

```

Datatrieve Procedure CON_STORE (cont.)

```

DECLARE TMSG4      PIC X(80).      0084
DECLARE TMSG5      PIC X(80).      0085
DECLARE TMSG6      PIC X(80).      0086
DECLARE TMSG7      PIC X(80).      0087
DECLARE TMSG8      PIC X(80).      0088
DECLARE TMSG9      PIC X(80).      0089
DECLARE TMSG10     PIC X(80).      0090
DECLARE TMSG10A    PIC X(80).      0091
DECLARE TMSG11     PIC X(80).      0092
!                               0093
!                               0094
!===== 0095
!  VARIABLES USED AS COUNTERS  0096
!===== 0097
!                               0098
!                               0099
!                               0100
DECLARE ICNT      PIC 9(4).      0101
DECLARE JCNT      PIC 9(4).      0102
DECLARE I         PIC 9(4).      0103
DECLARE J         PIC 9(4).      0104
!                               0105
!                               0106
!===== 0107
!  VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE  0108
!===== 0109
!                               0110
!                               0111
!                               0112
DECLARE CAL      USAGE DATE      0113
                  EDIT_STRING X(23). 0114
!                               0115
!                               0116
!===== 0117
!                               0118
!  READY THE APPROPRIATE DOMAINS, INITIALIZE THE MESSAGE VARIABLES AND  0119
!  INITIALIZE THE COUNTER (ICNT) USED FOR NUMBERING THE LOG FILE RECORDS 0120
!===== 0121
!                               0122
!                               0123
SET ABORT      0124
READY CONNECTIONS      SHARED WRITE 0125
READY SYSTEMS          SHARED READ   0126
READY MODULES          SHARED READ   0127
READY CONNECTIONS FORM SHARED READ   0128
TMSG1  = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 0129
TMSG2  = "SYSTEM A IS NOT VALID -- NOT IN DOMAIN SYSTEMS"          0130
TMSG3  = "MODULE A IS NOT VALID -- NOT IN DOMAIN MODULES FOR SYSTEM" 0131
TMSG4  = "CONNECTION IS NOT VALID -- NOT IN TABLE CONNECTION_TABLE" 0132
TMSG5  = "UNANTICIPATED CONNECTION IS NOT VALID -- MUST BE T OR F" 0133
TMSG6  = "UNANTICIPATED CONNECTION IS NOT VALID -- MUST BE F FOR CONNECTION" 0134

```

Datatrieve Procedure CON_STORE (cont.)

```

TMSG7   = "SYSTEM B IS NOT VALID -- NOT IN DOMAIN SYSTEMS"           0135
TMSG8   = "MODULE B IS NOT VALID -- NOT IN DOMAIN MODULES FOR SYSTEM" 0136
TMSG9   = "SYSTEM & MODULE B IS NOT VALID -- MUST NOT BE SAME AS SYSTEM & " | 0137
        "MODULE A"                                                    0138
TMSG10  = "CONNECTION"                                                0139
TMSG10A = "NOT VALID -- ALREADY IN DOMAIN CONNECTIONS"                0140
TMSG11  = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"                0141
ICNT    = 0                                                            0142
JCNT    = 0                                                            0143
!                                                                           0144
!                                                                           0145
!=====                                                                0146
! PRIMARY LOOP TO STORE FAILUREMODES                                   0147
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:                       0148
! 1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR         0149
!    CORRECTION OF INVALID INFORMATION                                0150
! 4. IF TCONTINUE NE "A", SECTION TO STORE RECORD IN DOMAIN           0151
!    CONNECTIONS AFTER VALIDATION TESTS HAVE BEEN PASSED             0152
! 5. IF TCONTINUE NE "A", SECTION TO PRINT DATA STORED IN DOMAIN     0153
!    CONNECTIONS FOR INCLUSION IN THE SESSION LOG FILE                0154
! 6. IF TCONTINUE = "A", SECTION TO PRINT MESSAGE THAT DATA CURRENTLY 0155
!    ON FORM HAS NOT BEEN STORED                                       0156
! 7. IF TCONTINUE = "A", SECTION TO REQUEST RESPONSE TO CONTINUE      0157
!    PROCEDURE OR EXIT TO MENU                                         0158
!=====                                                                0159
!                                                                           0160
!                                                                           0161
!                                                                           0162
!                                                                           0163
!                                                                           0164
TCONTINUE = "Y"                                                         0165
WHILE TCONTINUE = "Y"                                                    0166
BEGIN                                                                    0167
!                                                                           0168
!=====                                                                0169
!                                                                           0170
! LOOP TO DISPLAY BLANK TDMS FORMS, RETRIEVE THE DATA ENTERED ON THE 0171
! FORMS, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 0172
!=====                                                                0173
!                                                                           0174
!                                                                           0175
!                                                                           0176
FLG1 = "N"                                                                0177
IMSG = 1                                                                  0178
WHILE FLG1 = "N"                                                          0179
BEGIN                                                                    0180
    IF IMSG = 1 THEN TMSG = TMSG1                                       0181
    IF IMSG = 2 THEN TMSG = TMSG2                                       0182
    IF IMSG = 3 THEN TMSG = TMSG3 || " " | TSYSTEMA                    0183
    IF IMSG = 4 THEN TMSG = TMSG4                                       0184
    IF IMSG = 5 THEN TMSG = TMSG5                                       0185
    IF IMSG = 6 THEN TMSG = TMSG6 || " " | TCONNECT

```


Datatrieve Procedure CON_STORE (cont.)

```

IF IMMSG = 7 THEN TMSG = TMSG7                                0186
IF IMMSG = 8 THEN TMSG = TMSG8 || " " | TSYSTEMB              0187
IF IMMSG = 9 THEN TMSG = TMSG9                                0188
IF IMMSG = 10 THEN TMSG = TMSG10 || " " | TSYSTEMA | " " |    0189
    TTMODULEA | " " | TCONNECT | " " |                        0190
    TUCONNECT | " " | TSYSTEMB | " " |                        0191
    TTMODULEB | " " | TMSG10A                                  0192
IF IMMSG = 11 THEN TMSG = TMSG11                               0193
                                                                0194
                                                                0195
=====                                                       0196
THIS SECTION DISPLAYS THE STORE CONNECTIONS FORM AND          0197
RETRIEVES THE DATA ENTERED ON THE FORM                       0198
=====                                                       0199
                                                                0200
                                                                0201
                                                                0202
FOR FIRST 1 CONNECTIONS_FORM                                  0203
  BEGIN                                                       0204
    DISPLAY FORM CONNECTIONS STO FORM IN                       0205
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING               0206
    BEGIN                                                       0207
      IF IMMSG NE 1 THEN                                       0208
        BEGIN                                                 0209
          PUT FORM SYSTEM_A = TSYSTEMA                        0210
          PUT FORM MODULE_A = TMODULEA                        0211
          PUT FORM CONNECTION = TCONNECT                       0212
          PUT FORM UNANTICIPATED = TUCONNECT                   0213
          PUT FORM SYSTEM_B = TSYSTEMB                         0214
          PUT FORM MODULE_B = TMODULEB                         0215
        END                                                    0216
        PUT FORM CONTINUE = TCONTINUE                          0217
        PUT FORM MESSAGE = TMSG                                0218
      END RETRIEVE USING                                         0219
      BEGIN                                                     0220
        TSYSTEMA = GET FORM SYSTEM_A                           0221
        TMODULEA = GET FORM MODULE_A                           0222
        TCONNECT = GET FORM CONNECTION                          0223
        TUCONNECT = GET FORM UNANTICIPATED                     0224
        TSYSTEMB = GET FORM SYSTEM_B                           0225
        TMODULEB = GET FORM MODULE_B                           0226
        TCONTINUE = GET FORM CONTINUE                           0227
      END                                                       0228
    END                                                         0229
    TTMODULEA = TMODULEA                                         0230
    IF TMODULEA LT 1000 THEN TTMODULEA = "0" | TTMODULEA      0231
    IF TMODULEA LT 100 THEN TTMODULEA = "0" | TTMODULEA        0232
    IF TMODULEA LT 10 THEN TTMODULEA = "0" | TTMODULEA         0233
    TTMODULEB = TMODULEB                                         0234
    IF TMODULEB LT 1000 THEN TTMODULEB = "0" | TTMODULEB       0235
    IF TMODULEB LT 100 THEN TTMODULEB = "0" | TTMODULEB        0236

```

Datatrieve Procedure CON_STORE (cont.)

```

IF TMODULEB LT 10 THEN TTMODULEB = "0" | TTMODULEB
=====
IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
VERIFY THE FOLLOWING:
1. TSYSTEMA IS IN DOMAIN SYSTEMS
2. TMODULEA IS IN DOMAIN MODULES FOR SYSTEM = TSYSTEMA
3. TCONNECT IS IN TABLE CONNECTION_TABLE
4. TUCONNECT IS "T" OR "F"
5. TUCONNECT IS "F" FOR TCONNECT = "MECP" OR "MERE"
6. TSYSTEMB IS IN DOMAIN SYSTEMS
7. TMODULEB IS IN DOMAIN MODULES FOR SYSTEM = TSYSTEMB
8. TSYSTEMA & TMODULEA DO NOT EQUAL TSYSTEMB & TMODULEB
9. TSYSTEMA CONCATENATED WITH TTMODULEA IS LESS THAN
   TSYSTEMB CONCATENATED WITH TTMODULEB
10. TSYSTEMA, TMODULEA, TCONNECT, TUCONNECT, TSYSTEMB AND
    TMODULEB DO NOT ALREADY EXIST IN DOMAIN CONNECTIONS
11. TCONTINUE IS "Y", "N" OR "A"
=====
FLG1 = "Y"
IF TCONTINUE NE "A" THEN
  BEGIN
    FLG2 = "N"
    FOR SYSTEMS WITH SYSTEM = TSYSTEMA
      BEGIN
        FLG2 = "Y"
      END
    IF FLG2 = "N" THEN
      BEGIN
        FLG1 = "N"
        IMSG = 2
      END
    IF FLG1 = "Y" THEN
      BEGIN
        FLG2 = "N"
        FOR MODULES WITH SYSTEM_MODULE = TSYSTEMA | TTMODULEA
          BEGIN
            FLG2 = "Y"
          END
        IF FLG2 = "N" THEN
          BEGIN
            FLG1 = "N"
            IMSG = 3
          END
        IF FLG1 = "Y" THEN
          BEGIN

```

Datatrieve Procedure CON_STORE (cont.)

IF TCONNECT NOT IN CONNECTION_TABLE THEN	0288
BEGIN	0289
FLG1 = "N"	0290
IMSG = 4	0291
END	0292
END	0293
IF FLG1 = "Y" THEN	0294
BEGIN	0295
IF TUCONNECT NE "T" AND	0296
TUCONNECT NE "F" THEN	0297
BEGIN	0298
FLG1 = "N"	0299
IMSG = 5	0300
END	0301
END	0302
IF FLG1 = "Y" THEN	0303
BEGIN	0304
IF (TCONNECT = "MECP" OR	0305
TCONNECT = "MERE") AND	0306
TUCONNECT = "T" THEN	0307
BEGIN	0308
FLG1 = "N"	0309
IMSG = 6	0310
END	0311
END	0312
IF FLG1 = "Y" THEN	0313
BEGIN	0314
FLG2 = "N"	0315
FOR SYSTEMS WITH SYSTEM = TSYSTEMB	0316
BEGIN	0317
FLG2 = "Y"	0318
END	0319
IF FLG2 = "N" THEN	0320
BEGIN	0321
FLG1 = "N"	0322
IMSG = 7	0323
END	0324
END	0325
IF FLG1 = "Y" THEN	0326
BEGIN	0327
FLG2 = "N"	0328
FOR MODULES WITH SYSTEM_MODULE = TSYSTEMB TTMODULEB	0329
BEGIN	0330
FLG2 = "Y"	0331
END	0332
IF FLG2 = "N" THEN	0333
BEGIN	0334
FLG1 = "N"	0335
IMSG = 8	0336
END	0337
END	0338

Datatrieve Procedure CON_STORE (cont.)

```

IF FLG1 = "Y" THEN                                0339
  BEGIN                                            0340
    IF TSYSTEMA | TTMODULEA = TSYSTEMB | TTMODULEB THEN 0341
      BEGIN                                        0342
        FLG1 = "N"                                0343
        IMSG = 9                                  0344
      END                                          0345
    END                                          0346
  IF FLG1 = "Y" THEN                                0347
    BEGIN                                        0348
      IF TSYSTEMA | TTMODULEA GT TSYSTEMB | TTMODULEB THEN 0349
        BEGIN                                        0350
          TSYSTEM = TSYSTEMA                        0351
          TMODULE = TMODULEA                        0352
          TTMODULE = TTMODULEA                      0353
          TSYSTEMA = TSYSTEMB                       0354
          TMODULEA = TMODULEB                       0355
          TTMODULEA = TTMODULEB                     0356
          TSYSTEMB = TSYSTEM                        0357
          TMODULEB = TMODULE                        0358
          TTMODULEB = TTMODULE                      0359
        END                                          0360
      END                                          0361
    IF FLG1 = "Y" THEN                                0362
      BEGIN                                        0363
        FOR CONNECTIONS WITH CODE_NUMBER = TSYSTEMA | TTMODULEA | 0364
          TCONNECT | TUCONNECT | 0365
          TSYSTEMB | TTMODULEB 0366
        BEGIN                                        0367
          FLG1 = "N"                                0368
          IMSG = 10                                  0369
        END                                          0370
      END                                          0371
    IF FLG1 = "Y" THEN                                0372
      BEGIN                                        0373
        IF TCONTINUE NE "Y" AND                      0374
          TCONTINUE NE "N" AND                      0375
          TCONTINUE NE "A" THEN                    0376
          BEGIN                                        0377
            FLG1 = "N"                                0378
            IMSG = 11                                  0379
          END                                          0380
        END                                          0381
      END                                          0382
    END                                          0383
  END                                          0384
END                                          0385
===== 0386
! IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN 0387
! DOMAIN CONNECTIONS 0388
! 0389

```

Datatrieve Procedure CON_STORE (cont.)

```

=====
IF TCONTINUE NE "A" THEN
  BEGIN
    CAL = "NOW"
    STORE CONNECTIONS USING
      BEGIN
        DATE_CREATED          = CAL
        SYSTEM_A              = TSYSTEMA
        MODULE_A              = TMODULEA
        CONNECTION            = TCONNECT
        UNANTICIPATED_CONNECTION = TUCONNECT
        SYSTEM_B              = TSYSTEMB
        MODULE_B              = TMODULEB
      END
    END
  END
=====
IF TCONTINUE IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN
CONNECTIONS IS PRINTED.  THE OUTPUT OF THE PRINT STATEMENTS WILL
BE INCLUDED IN THE SESSION LOG FILE WHICH IS OPENED BY THE CALLING
COMMAND PROCEDURE.
=====
IF TCONTINUE NE "A" THEN
  BEGIN
    ICNT = ICNT + 1
    JCNT = JCNT + 1
    IF JCNT = 1 THEN PRINT NEW PAGE
    FOR CONNECTIONS WITH CODE_NUMBER = TSYSTEMA | TTMODULEA |
                                     TCONNECT  | TUCONNECT  |
                                     TSYSTEMB  | TTMODULEB  |
    PRINT SKIP 3,
      COL 1, "RECORD NO.",          SPACE 1,
            ICNT (-) USING ZZ9, SKIP 1,
      COL 1, "=====", SPACE 0,
            "=====", SPACE 0,
            "=====", SPACE 0,
            "=====", SKIP 2,
      COL 3, "DATE_CREATED          :", SPACE 1,
            DATE_CREATED (-) USING X(23), SKIP 1,
      COL 3, "SYSTEM MODULE A      :", SPACE 1,
            SYSTEM_A (-) USING X(4),      SPACE 1,
            MODULE_A (-) USING 9(4),      SKIP 1,
      COL 3, "CONNECTION           :", SPACE 1,
            CONNECTION (-) USING X(4),    SKIP 1,
      COL 3, "UNANTICIPATED_CONNECTION :",          SPACE 1,

```

Datatrieve Procedure CON_STORE (cont.)

```

                                UNANTICIPATED CONNECTION (-) USING X(1), SKIP 1,      0441
COL 3, "SYSTEM MODULE B      :", SPACE 1,      0442
                                SYSTEM_B (-) USING X(4),      SPACE 1,      0443
                                MODULE_B (-) USING 9(4),      SKIP 2,      0444
COL 1, "===== ", SPACE 0,      0445
                                "===== ", SPACE 0,      0446
                                "===== ", SPACE 0,      0447
                                "===== "      0448
IF JCNT = 4 THEN JCNT = 0      0449
:BELL      0450
END      0451
!      0452
!      0453
!      0454
!      0455
!      0456
!      0457
!      0458
!      0459
!      0460
!      0461
!      0462
!      0463
!      0464
!      0465
!      0466
!      0467
!      0468
!      0469
!      0470
!      0471
!      0472
!      0473
!      0474
!      0475
!      0476
!      0477
!      0478
!      0479
!      0480
!      0481
!      0482
!      0483
!      0484
!      0485
!      0486
!      0487
!      0488
!      0489
!      0490
!      0491

IF TCONTINUE IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED. THIS
MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE.

=====
IF TCONTINUE = "A" THEN
BEGIN
    JCNT = JCNT + 1
    IF JCNT = 1 THEN PRINT NEW_PAGE
    PRINT SKIP 3,
        COL 1, "===== ", SPACE 0,
        "===== ", SKIP 2,
        COL 9, "*****", SKIP 2,
        COL 9, "***** RECORD NOT STORED *****", SKIP 2,
        COL 9, "*****", SKIP 2,
        COL 1, "===== ", SPACE 0,
        "===== "
    IF JCNT = 4 THEN JCNT = 0
END

=====
IF TCONTINUE EQUALS "A", A RESPONSE IS REQUESTED TO EITHER CONTINUE
THE PROCEDURE TO STORE CONNECTIONS OR EXIT TO THE MENU

=====
IF TCONTINUE = "A" THEN
BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    TCONTINUE = "X"
    WHILE TCONTINUE NE "Y" AND
        TCONTINUE NE "N"

```

Datatrieve Procedure CON_STORE (cont.)

BEGIN	0492
PRINT SKIP 2,	0493
"Do you wish to continue entering CONNECTIONS?",	0494
SKIP 1	0495
TCONTINUE = FN\$UPCASE(*."Y or N")	0496
PRINT " "	0497
END	0498
END	0499
END	0500
END-PROCEDURE	0501

Datatrieve Procedure CREATE_CONNECTIONS

DEFINE PROCEDURE CREATE_CONNECTIONS	0502
DEFINE FILE FOR CONNECTIONS KEY = DATE_CREATED (DUP),	0503
KEY = CODE_NUMBER	0504
END-PROCEDURE	0505

Datatrieve Procedure CREATE_CONNECTIONS_FORM

DEFINE PROCEDURE CREATE CONNECTIONS FORM	0506
DEFINE FILE FOR CONNECTIONS_FORM KEY = DATE_CREATED (DUP),	0507
KEY = CODE_NUMBER	0508
END-PROCEDURE	0509

Datatrieve Procedure CREATE_FAILUREMODES

DEFINE PROCEDURE CREATE_FAILUREMODES	0510
DEFINE FILE FOR FAILUREMODES KEY = DATE_CREATED (DUP),	0511
KEY = FMCODE	0512
END-PROCEDURE	0513

Datatrieve Procedure CREATE_FAILUREMODES_FORM

DEFINE PROCEDURE CREATE_FAILUREMODES_FORM	0514
DEFINE FILE FOR FAILUREMODES_FORM KEY = DATE_CREATED (DUP),	0515
KEY = FMCODE	0516
END-PROCEDURE	0517

Datatrieve Procedure CREATE_MODULES

DEFINE PROCEDURE CREATE_MODULES	0518
DEFINE FILE FOR MODULES KEY = DATE_CREATED (DUP),	0519
KEY = SYSTEM_MODULE	0520
KEY = SYSTEM_MODULE_NAME (DUP)	0521
END-PROCEDURE	0522

Datatrieve Procedure CREATE_MODULES_FORM

DEFINE PROCEDURE CREATE_MODULES_FORM	0523
DEFINE FILE FOR MODULES_FORM KEY = DATE_CREATED (DUP),	0524
KEY = SYSTEM_MODULE	0525
KEY = SYSTEM_MODULE_NAME (DUP)	0526
END-PROCEDURE	0527

Datatrieve Procedure CREATE_PROPAGATIONS

DEFINE PROCEDURE CREATE_PROPAGATIONS	0528
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),	0529
KEY = FMCODE (DUP),	0530
KEY = CODE_NUMBER (DUP),	0531
KEY = SIGNAL_TYPE (DUP)	0532
END-PROCEDURE	0533

[illegible]

```

DEFINE PROCEDURE CREATE_PROPAGATIONS_FIP_2
!
!
=====
!
!  PROCEDURE TO CREATE THE PROPAGATIONS DOMAIN AND DATA FILE FOR THE
!  CURRENT SYSTEM.  DATATRIEVE LOGICALS ARE CREATED TO REPRESENT THE
!  NAMES OF THE DOMAIN AND ASSOCIATED DATA FILE.  THE DOMAIN IS NAMED
!  "PROPAGATIONS_X999" WHERE "X999" IS THE CURRENT SYSTEM.  THE DATA FILE
!  IS NAMED "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_X999".  AFTER
!  CREATING THE DOMAIN AND DATA FILE, PROCEDURE CREATE_PROPAGATIONS_FIP_1
!  IS TO SEARCH FOR ADDITIONAL SYSTEMS.
!
=====
!
FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_" | TSYSTEM)
FN$CREATE_LOG("PROPAGATIONS_FILE",
"DEV$206:[BCDSSME2.DATA]PROPAGATIONS_" | TSYSTEM | ".DAT")
DEFINE DOMAIN PROPAGATIONS USING PROPAGATIONS_REC ON PROPAGATIONS_FILE;
DEFINE FILE FOR PROPAGATIONS KEY = DATE_CREATED (DUP),
KEY = FMCODE (DUP),
KEY = CODE_NUMBER (DUP),
KEY = SIGNAL_TYPE (DUP)
FOR SYSTEMS WITH SYSTEM = TSYSTEM
BEGIN
CAL = "NOW"
MODIFY USING
BEGIN
PROPAGATIONS_FILE_CREATED = "YES"
DATE_LAST_MODIFIED = CAL
MODIFYING_PROCEDURE = "FIP_STORE"
END
END
FN$DELETE_LOG("PROPAGATIONS")
FN$DELETE_LOG("PROPAGATIONS_FILE")
:CREATE_PROPAGATIONS_FIP_1
END-PROCEDURE

```


Datatrieve Procedure CREATE_PROPAGATIONS_FORM

DEFINE PROCEDURE CREATE_PROPAGATIONS_FORM	0597
DEFINE FILE FOR PROPAGATIONS_FORM KEY = DATE_CREATED (DUP),	0598
KEY = FMCODE (DUP),	0599
KEY = CODE_NUMBER (DUP),	0600
KEY = SIGNAL_TYPE (DUP)	0601
END-PROCEDURE	0602

Datatrieve Procedure CREATE_PROPAGATIONS_SYS_1

[illegible]

[illegible]

Datatrieve Procedure CREATE_REFERENCES

DEFINE PROCEDURE CREATE_REFERENCES		0666
DEFINE FILE FOR REFERENCES	KEY = DATE_CREATED (DUP),	0667
	KEY = REFERENCE_NUMBER ,	0668
	KEY = DOCUMENT_TITLE (DUP),	0669
	KEY = DOCUMENT_SOURCE (DUP)	0670
END-PROCEDURE		0671

Datatrieve Procedure CREATE_REFERENCES_FORM

DEFINE PROCEDURE CREATE_REFERENCES_FORM	0672
DEFINE FILE FOR REFERENCES_FORM KEY = DATE_CREATED (DUP),	0673
KEY = REFERENCE_NUMBER ,	0674
KEY = DOCUMENT_TITLE (DUP),	0675
KEY = DOCUMENT_SOURCE (DUP)	0676
END-PROCEDURE	0677

Datatrieve Procedure CREATE_SYSTEMS

DEFINE PROCEDURE CREATE_SYSTEMS	0678
DEFINE FILE FOR SYSTEMS KEY = DATE_CREATED (DUP),	0679
KEY = SYSTEM	0680
KEY = SYSTEM_NAME	0681
END-PROCEDURE	0682

Datatrieve Procedure CREATE_SYSTEMS_FORM

DEFINE PROCEDURE CREATE_SYSTEMS_FORM	0683
DEFINE FILE FOR SYSTEMS_FORM KEY = DATE_CREATED (DUP),	0684
KEY = SYSTEM	0685
KEY = SYSTEM_NAME	0686
END-PROCEDURE	0687

Datatrieve Procedure DTR_NULL

```
DEFINE PROCEDURE DTR_NULL                                0688
!                                                         0689
!                                                         0690
!=====                                                 0691
!                                                         0692
! THIS PROCEDURE IS EMPTY. IT CONTAINS NO EXECUTABLE DATATRIEVE 0693
! COMMANDS OR STATEMENTS. IT IS USED TO TERMINATE PROCEDURES WHICH 0694
! CONTAIN DATATRIEVE COMMANDS AND LOOP BY SUCCESSIVE CALLS TO THE SAME 0695
! PROCEDURE.                                             0696
!=====                                                 0697
!                                                         0698
!                                                         0699
END-PROCEDURE                                           0700
```


Datatrieve Procedure FIPLOGICALC

DEFINE PROCEDURE FIPLOGICALC	0701
DECLARE SYSTEM PIC X(4).	0702
:CLRSCRN	0703
PRINT SKIP 4,	0704
COL 1, "===== DEFINITION OF DTR LOGICALS FOR PROPAGATIONS AND", SPACE 1,	0705
"PROPAGATIONS FILE =====", SKIP 2	0706
SYSTEM = FN\$UPCASE(*."SYSTEM")	0707
IF FN\$TRANS LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN	0708
FN\$DELETE LOG("PROPAGATIONS")	0709
FN\$CREATE LOG("PROPAGATIONS", "PROPAGATIONS_" SYSTEM)	0710
IF FN\$TRANS LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN	0711
FN\$DELETE LOG("PROPAGATIONS_FILE")	0712
FN\$CREATE LOG("PROPAGATIONS_FILE",	0713
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_" SYSTEM ".DAT")	0714
END-PROCEDURE	0715

Datatrieve Procedure FIPLOGICALD

DEFINE PROCEDURE FIPLOGICALD	0716
:CLRSCRN	0717
PRINT SKIP 4,	0718
COL 1, "===== DELETION OF DTR LOGICALS FOR PROPAGATIONS AND", SPACE 1,	0719
"PROPAGATIONS_FILE =====", SKIP 2	0720
FN\$DELETE_LOG("PROPAGATIONS")	0721
FN\$DELETE_LOG("PROPAGATIONS_FILE")	0722
END-PROCEDURE	0723

Datatrieve Procedure FIP_COUNT

DEFINE PROCEDURE FIP_COUNT	0724
DECLARE ICNT PIC 9(5).	0725
DECLARE JCNT PIC 9(5).	0726
DECLARE KCNT PIC 9(5).	0727
DECLARE LCNT PIC 9(5).	0728
DECLARE MCNT PIC 9(5).	0729
DECLARE NCNT PIC 9(5).	0730
DECLARE PCNT PIC 9(5).	0731
DECLARE HSYSTEM PIC X(4).	0732
READY SYSTEMS SHARED READ	0733
ICNT = 0	0734
FOR SYSTEMS WITH PROPAGATIONS_FILE_CREATED = "YES"	0735
ICNT = ICNT + 1	0736
JCNT = 0	0737
MCNT = 0	0738
:CLRSCRN	0739
PRINT "The number of Failure Information Propagation", SKIP 1,	0740
"records are as follows:", SKIP 1	0741
:FIP COUNT 1	0742
PRINT SKIP 1,	0743
COL 7, "TOTAL",	0744
COL 30, MCNT (-) USING ZZZZ9	0745
NCNT = MCNT / 2	0746
PCNT = NCNT * 2	0747
IF PCNT LT MCNT THEN PCNT = PCNT + 1	0748
PRINT SKIP 1,	0749
"With 2 records per page the resulting file", SKIP 1,	0750
"will contain", SPACE 1, NCNT (-) USING ZZZZ9, SPACE 1,	0751
"pages."	0752
IF FN\$TRANS LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN	0753
FN\$DELETE LOG("PROPAGATIONS")	0754
IF FN\$TRANS LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN	0755
FN\$DELETE LOG("PROPAGATIONS_FILE")	0756
IF FN\$TRANS LOG("PROC") NE "PROC" THEN	0757
FN\$DELETE LOG("PROC")	0758
END-PROCEDURE	0759

Datatrieve Procedure FIP_COUNT_1

```

DEFINE PROCEDURE FIP_COUNT_1                                0760
JCNT = JCNT + 1                                             0761
IF JCNT LE ICNT THEN                                       0762
  BEGIN                                                    0763
    KCNT = 0                                               0764
    FOR SYSTEMS WITH                                       0765
      PROPAGATIONS_FILE_CREATED = "YES" SORTED BY        0766
      SYSTEM                                               0767
      BEGIN                                                0768
        KCNT = KCNT + 1                                     0769
        IF KCNT = JCNT THEN                                0770
          HSYSTEM = SYSTEM                                  0771
        END                                                0772
        IF FN$TRANS LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN 0773
          FN$DELETE LOG("PROPAGATIONS")                    0774
        IF FN$TRANS LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN 0775
          FN$DELETE LOG("PROPAGATIONS_FILE")              0776
        IF FN$TRANS LOG("PROC") NE "PROC" THEN            0777
          FN$DELETE LOG("PROC")                            0778
        FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_" | HSYSTEM) 0779
        FN$CREATE_LOG("PROPAGATIONS_FILE",                0780
          "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_" | HSYSTEM | ".DAT") 0781
        FN$CREATE_LOG("PROC", "FIP_COUNT_2")              0782
      END                                                    0783
    IF JCNT GT ICNT THEN                                    0784
      BEGIN                                                0785
        IF FN$TRANS LOG("PROC") NE "PROC" THEN            0786
          FN$DELETE LOG("PROC")                            0787
        FN$CREATE_LOG("PROC", "DTR_NULL")                  0788
      END                                                    0789
    :PROC                                                  0790
  END-PROCEDURE                                           0791

```

Datatrieve Procedure FIP_COUNT_2

DEFINE PROCEDURE FIP_COUNT_2	0792
READY PROPAGATIONS SHARED READ	0793
LCNT = 0	0794
FOR PROPAGATIONS LCNT = LCNT + 1	0795
PRINT COL 10, FN\$TRANS LOG("PROPAGATIONS") (-) USING X(17), SPACE 1,	0796
"=", SPACE 1, LCNT (-) USING ZZZZ9	0797
MCNT = MCNT + LCNT	0798
:FIP_COUNT_1	0799
END-PROCEDURE	0800

Datatrieve Procedure FIP_LIST_1

```

DEFINE PROCEDURE FIP_LIST_1                                0801
JCNT = JCNT + 1                                           0802
IF JCNT LE ICNT THEN                                     0803
  BEGIN                                                  0804
    KCNT = 0                                             0805
    FOR SYSTEMS WITH                                     0806
      PROPAGATIONS_FILE_CREATED = "YES" SORTED BY      0807
      SYSTEM                                             0808
      BEGIN                                             0809
        KCNT = KCNT + 1                                0810
        IF KCNT = JCNT THEN                             0811
          HSYSTEM = SYSTEM                             0812
        END                                             0813
        IF FN$TRANS_LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN 0814
          FN$DELETE_LOG("PROPAGATIONS")                 0815
        IF FN$TRANS_LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN 0816
          FN$DELETE_LOG("PROPAGATIONS_FILE")            0817
        IF FN$TRANS_LOG("PROC") NE "PROC" THEN          0818
          FN$DELETE_LOG("PROC")                         0819
        FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_" | HSYSTEM) 0820
        FN$CREATE_LOG("PROPAGATIONS_FILE",              0821
          "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_" | HSYSTEM | ".DAT") 0822
        FN$CREATE_LOG("PROC", "FIP_LIST_2")             0823
      END                                               0824
    IF JCNT GT ICNT THEN                                0825
      BEGIN                                             0826
        IF FN$TRANS_LOG("PROC") NE "PROC" THEN        0827
          FN$DELETE_LOG("PROC")                       0828
          FN$CREATE_LOG("PROC", "DTR_NULL")            0829
        END                                             0830
      :PROC                                           0831
    END-PROCEDURE                                     0832

```

Datatrieve Procedure FIP_LIST_2

DEFINE PROCEDURE FIP_LIST_2	0833
READY PROPAGATIONS SHARED READ	0834
LCNT = 0	0835
FOR PROPAGATIONS LCNT = LCNT + 1	0836
MCNT = MCNT + LCNT	0837
:FIP_LIST_1	0838
END-PROCEDURE	0839

Datatrieve Procedure FIP_LIST_3

```

DEFINE PROCEDURE FIP_LIST_3                                0840
JCNT = JCNT + 1                                           0841
IF JCNT LE ICNT THEN                                     0842
  BEGIN                                                  0843
    KCNT = 0                                             0844
    FOR SYSTEMS WITH                                     0845
      PROPAGATIONS_FILE_CREATED = "YES" SORTED BY      0846
      SYSTEM                                             0847
      BEGIN                                             0848
        KCNT = KCNT + 1                                 0849
        IF KCNT = JCNT THEN                             0850
          HSYSTEM = SYSTEM                             0851
        END                                             0852
        IF FN$TRANS_LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN 0853
          FN$DELETE_LOG("PROPAGATIONS")                 0854
        IF FN$TRANS_LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN 0855
          FN$DELETE_LOG("PROPAGATIONS_FILE")            0856
        IF FN$TRANS_LOG("PROPAGATIONS_LIST") NE "PROPAGATIONS_LIST" THEN 0857
          FN$DELETE_LOG("PROPAGATIONS_LIST")            0858
        IF FN$TRANS_LOG("PROC") NE "PROC" THEN          0859
          FN$DELETE_LOG("PROC")                         0860
        FN$CREATE_LOG("PROPAGATIONS", "PROPAGATIONS_" | HSYSTEM) 0861
        FN$CREATE_LOG("PROPAGATIONS_FILE",              0862
          "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_" | HSYSTEM | ".DAT") 0863
        FN$CREATE_LOG("PROPAGATIONS_LIST",              0864
          "DEV$206:[BCDSSME2.DATA]FIP_" | HSYSTEM | ".LST") 0865
        FN$CREATE_LOG("PROC", "FIP_LIST_4")             0866
      END                                             0867
    IF JCNT GT ICNT THEN                                0868
      BEGIN                                             0869
        IF FN$TRANS_LOG("PROC") NE "PROC" THEN        0870
          FN$DELETE_LOG("PROC")                        0871
        FN$CREATE_LOG("PROC", "DTR_NULL")              0872
      END                                             0873
    :PROC                                              0874
  END-PROCEDURE                                       0875

```


Datatrieve Procedure FIP_LIST_4

```

DEFINE PROCEDURE FIP_LIST_4                                0876
READY PROPAGATIONS SHARED READ                            0877
ON PROPAGATIONS_LIST                                     0878
BEGIN                                                      0879
    PCNT = 0                                                0880
    FOR PROPAGATIONS SORTED BY                             0881
        FMCODE,                                           0882
        SIGNAL_TYPE,                                     0883
        PARAMETER,                                       0884
        DATE_CREATED                                     0885
    BEGIN                                                  0886
        NCNT = NCNT + 1                                    0887
        PCNT = PCNT + 1                                    0888
        IF PCNT = 1 THEN                                  0889
            PRINT NEW_PAGE,                                0890
                COL 1,                                     0891
                "Domain", SPACE 1,                         0892
                FN$TRANS_LOG("PROPAGATIONS") (-) USING X(17), 0893
                COL 63, CAL (-) USING X(17)                0894
        PRINT SKIP 1,                                     0895
            COL 1, "RECORD NO.", SPACE 1,                 0896
                NCNT (-) USING ZZZ9, SPACE 1,             0897
                "OF", SPACE 1,                             0898
                MCNT (-) USING ZZZ9, SKIP 1,              0899
            COL 1, "=====",                             0900
                SPACE 0,                                   0901
                "===== ",                               0902
                SKIP 2,                                    0903
            COL 1, "DATE_CREATED :", SPACE 1,             0904
                DATE_CREATED (-) USING X(23), SKIP 1,     0905
            COL 1, "FMCODE :", SPACE 1,                   0906
                SOURCE_SYSTEM (-) USING X(4), SPACE 1,    0907
                SOURCE_MODULE (-) USING 9(4), SPACE 1,    0908
                FAILURE_MODE (-) USING X(2), SPACE 1,     0909
                FAILURE_SUBMODE (-) USING X(2), SPACE 1,  0910
                ACCOMPLICE_SYSTEM (-) USING X(4), SPACE 1, 0911
                ACCOMPLICE_MODULE (-) USING 9(4), SKIP 1,  0912
            COL 1, "CODE_NUMBER :", SPACE 1,              0913
                SYSTEM_A (-) USING X(4), SPACE 1,        0914
                MODULE_A (-) USING 9(4), SPACE 1,        0915
                CONNECTION_TYPE (-) USING X(2), SPACE 1,  0916
                CONNECTION_QUALIFIER (-) USING X(2), SPACE 1, 0917
                UNANTICIPATED_CONNECTION (-) USING X(1), SPACE 1, 0918
                SYSTEM_B (-) USING X(4), SPACE 1,        0919
                MODULE_B (-) USING 9(4), SKIP 1,          0920
            COL 1, "SIGNAL_TYPE :", SPACE 1,             0921
                SIGNAL_TYPE (-) USING X(20), SKIP 1,     0922
            COL 1, "SIGNAL_UNITS :", SPACE 1,            0923
                SIGNAL_UNITS (-) USING X(25), SKIP 1,    0924
            COL 1, "DIMENSIONS :", SPACE 1,              0925
                DIMENSIONS (-) USING 9(1), SKIP 1,        0926
    END

```

Datatrieve Procedure FIP_LIST_4 (cont.)

COL 1, "SIGNAL_QUALITY :	SPACE 1,	0927
SIGNAL_QUALITY (-) USING 9(1),	SKIP 1,	0928
COL 1, "MAX_FREQ_OR_TIME :	SPACE 1,	0929
MAX_FREQ_OR_TIME (-) USING -Z9,	SKIP 1,	0930
COL 1, "MIN_FREQ_OR_TIME :	SPACE 1,	0931
MIN_FREQ_OR_TIME (-) USING -Z9,	SKIP 1,	0932
COL 1, "FT_UNITS :	SPACE 1,	0933
FT_UNITS (-) USING X(25),	SKIP 1,	0934
COL 1, "PARAMETER :	SPACE 1,	0935
PARAMETER (-) USING X(20),	SKIP 1,	0936
COL 1, "PARAMETER_UNITS :	SPACE 1,	0937
PARAMETER_UNITS (-) USING X(25),	SKIP 1,	0938
COL 1, "SYMPTOM_DURATION :	SPACE 1,	0939
SYMPTOM_DURATION (-) USING -Z9,	SKIP 1,	0940
COL 1, "PERIOD_OF_ONSET :	SPACE 1,	0941
PERIOD_OF_ONSET (-) USING -Z9,	SKIP 1,	0942
COL 1, "INDICATES_FAILURE :	SPACE 1,	0943
INDICATES_FAILURE (-) USING X(1),	SKIP 1,	0944
COL 1, "COMMENT1 :	SPACE 1,	0945
COMMENT1 (-) USING T(58),	SKIP 1,	0946
COL 1, "COMMENT2 :	SPACE 1,	0947
COMMENT2 (-) USING T(58),	SKIP 1,	0948
COL 1, "COMMENT3 :	SPACE 1,	0949
COMMENT3 (-) USING T(58),	SKIP 1,	0950
COL 1, "DATE_LAST_MODIFIED :	SPACE 1,	0951
DATE_LAST_MODIFIED (-) USING X(23),	SKIP 1,	0952
COL 1, "MODIFYING_PROCEDURE :	SPACE 1,	0953
MODIFYING_PROCEDURE (-) USING X(20),	SKIP 2,	0954
COL 1, "=====		0955
SPACE 0,		0956
"=====		0957
IF PCNT = 2 THEN PCNT = 0		0958
END		0959
END		0960
:FIP_LIST 3		0961
END-PROCEDURE		0962

Datatrieve Procedure FIP_MODIFY

DEFINE PROCEDURE FIP_MODIFY	0963
!	0964
!	0965
=====	0966
!	0967
! VARIABLE ASSOCIATED WITH SYSTEM FOR WHICH FAILURE INFORMATION	0968
! PROPAGATION DATA IS CURRENTLY BEING MODIFIED	0969
!	0970
=====	0971
!	0972
DECLARE TSYSTEM PIC X(4).	0973
!	0974
!	0975
=====	0976
!	0977
! VARIABLES ASSOCIATED WITH FMCODE	0978
!	0979
=====	0980
!	0981
DECLARE TFMCODE PIC X(20).	0982
DECLARE TSSYS PIC X(4).	0983
!	0984
!	0985
=====	0986
!	0987
! VARIABLES ASSOCIATED WITH CODE_NUMBER	0988
!	0989
=====	0990
!	0991
DECLARE TCODENO PIC X(21).	0992
DECLARE TSYSA PIC X(4).	0993
DECLARE TSYSMODA PIC X(8).	0994
DECLARE TCONNECTU PIC X(5).	0995
DECLARE TSYSB PIC X(4).	0996
DECLARE TSYSMODB PIC X(8).	0997
!	0998
!	0999
=====	1000
!	1001
! VARIABLES ASSOCIATED WITH OTHER FIELDS FOR	1002
! DOMAIN PROPAGATIONS:	1003
! 1. SIGNAL_TYPE	1004
! 2. DIMENSIONS	1005
! 3. SIGNAL_QUALITY	1006
! 4. MAX_FREQ_OR_TIME	1007
! 5. MIN_FREQ_OR_TIME	1008
! 6. PARAMETER	1009
! 7. SYMPTOM_DURATION	1010
! 8. PERIOD_OF_ONSET	1011
! 9. INDICATES_FAILURE	1012
! 10. COMMENT1	1013

Datatrieve Procedure FIP_MODIFY (cont.)

!	11. COMMENT2	1014
!	12. COMMENT3	1015
!		1016
!=====		1017
!		1018
DECLARE	TSIG PIC X(20).	1019
DECLARE	TDIM PIC X(1).	1020
DECLARE	NDIM PIC 9(1).	1021
DECLARE	TSIGQUAL PIC X(1).	1022
DECLARE	NSIGQUAL PIC 9(1).	1023
DECLARE	TMAXFT PIC X(3).	1024
DECLARE	TMAXFT1 PIC X(1).	1025
DECLARE	TMAXFT2 PIC X(1).	1026
DECLARE	TMAXFT3 PIC X(1).	1027
DECLARE	NMAXFT PIC S9(2).	1028
DECLARE	TMINFT PIC X(3).	1029
DECLARE	TMINFT1 PIC X(1).	1030
DECLARE	TMINFT2 PIC X(1).	1031
DECLARE	TMINFT3 PIC X(1).	1032
DECLARE	NMINFT PIC S9(2).	1033
DECLARE	TPAR PIC X(20).	1034
DECLARE	TSYMDUR PIC X(3).	1035
DECLARE	TSYMDUR1 PIC X(1).	1036
DECLARE	TSYMDUR2 PIC X(1).	1037
DECLARE	TSYMDUR3 PIC X(1).	1038
DECLARE	NSYMDUR PIC S9(2).	1039
DECLARE	TPDONSET PIC X(3).	1040
DECLARE	TPDONSET1 PIC X(1).	1041
DECLARE	TPDONSET2 PIC X(1).	1042
DECLARE	TPDONSET3 PIC X(1).	1043
DECLARE	NPDONSET PIC S9(2).	1044
DECLARE	TINDFAIL PIC X(1).	1045
DECLARE	TCOMMENT1 PIC X(80).	1046
DECLARE	TCOMMENT2 PIC X(80).	1047
DECLARE	TCOMMENT3 PIC X(80).	1048
DECLARE	TCOMMENT PIC X(80).	1049
DECLARE	TTDIM PIC 9(1).	1050
DECLARE	TTSIGQUAL PIC 9(1).	1051
DECLARE	TTMAXFT PIC S9(2).	1052
DECLARE	TTMINFT PIC S9(2).	1053
DECLARE	TTSYMDUR PIC S9(2).	1054
DECLARE	TTPDONSET PIC S9(2).	1055
DECLARE	TTINDFAIL PIC X(1).	1056
DECLARE	TTCOMMENT1 PIC X(80).	1057
DECLARE	TTCOMMENT2 PIC X(80).	1058
DECLARE	TTCOMMENT3 PIC X(80).	1059
!		1060
!		1061
!=====		1062
!		1063
!	VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR COMMENTS	1064

Datatrieve Procedure FIP_MODIFY (cont.)

!	1065
=====	1066
!	1067
DECLARE TEMP PIC X(249).	1068
!	1069
!	1070
=====	1071
!	1072
! VARIABLES USED AS FLAGS OR CONDITION INDICATORS	1073
!	1074
=====	1075
!	1076
DECLARE TCONTINUE1 PIC X(1).	1077
DECLARE TCONTINUE2 PIC X(1).	1078
DECLARE TCONTINUE3 PIC X(1).	1079
DECLARE TCONTINUE4 PIC X(1).	1080
DECLARE TCONTINUE5 PIC X(1).	1081
DECLARE MSG PIC 9(2).	1082
DECLARE FLG1 PIC X(1).	1083
DECLARE FLG2 PIC X(1).	1084
DECLARE FLG3 PIC X(1).	1085
DECLARE FLG4 PIC X(1).	1086
DECLARE FLG5 PIC X(1).	1087
DECLARE FLG6 PIC X(1).	1088
!	1089
!	1090
=====	1091
!	1092
! VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORMS	1093
!	1094
=====	1095
!	1096
DECLARE TMSG PIC X(80).	1097
DECLARE TMSG1 PIC X(80).	1098
DECLARE TMSG2 PIC X(80).	1099
DECLARE TMSG3 PIC X(80).	1100
DECLARE TMSG4 PIC X(80).	1101
DECLARE TMSG5 PIC X(80).	1102
DECLARE TMSG6 PIC X(80).	1103
DECLARE TMSG7 PIC X(80).	1104
DECLARE TMSG8 PIC X(80).	1105
DECLARE TMSG9 PIC X(80).	1106
DECLARE TMSG10 PIC X(80).	1107
DECLARE TMSG11 PIC X(80).	1108
DECLARE TMSG12 PIC X(80).	1109
DECLARE TMSG13 PIC X(80).	1110
DECLARE TMSG14 PIC X(80).	1111
DECLARE TMSG15 PIC X(80).	1112
DECLARE TMSG16 PIC X(80).	1113
DECLARE TMSG17 PIC X(80).	1114
DECLARE TMSG18 PIC X(80).	1115

DECLARE TMSG19	PIC X(80).	1116
!		1117
!		1118
=====		1119
!		1120
! VARIABLES USED AS COUNTERS		1121
!		1122
=====		1123
!		1124
DECLARE ICNT	PIC 9(4).	1125
DECLARE JCNT	PIC 9(4).	1126
DECLARE KCNT	PIC 9(4).	1127
DECLARE I	PIC 9(4).	1128
DECLARE J	PIC 9(4).	1129
!		1130
!		1131
=====		1132
!		1133
! VARIABLES USED TO TEMPORARILY STORE FAILURE INFORMATION PROPAGATION		1134
! DATA FOR COMPARISON OF INITIAL AND MODIFIED VALUES		1135
!		1136
=====		1137
!		1138
DECLARE HDCREATED	USAGE DATE	1139
	EDIT STRING X(23).	1140
DECLARE HFMCODE	PIC X(20).	1141
DECLARE HCODENO	PIC X(21).	1142
DECLARE HSIG	PIC X(20).	1143
DECLARE HSIGUNIT	PIC X(25).	1144
DECLARE HDIM	PIC 9(1).	1145
DECLARE HSIGQUAL	PIC 9(1).	1146
DECLARE HMAXFT	PIC S9(2).	1147
DECLARE HMINFT	PIC S9(2).	1148
DECLARE HFTUNIT	PIC X(25).	1149
DECLARE HPAR	PIC X(20).	1150
DECLARE HPARUNIT	PIC X(25).	1151
DECLARE HSYMDUR	PIC S9(2).	1152
DECLARE HONSET	PIC S9(2).	1153
DECLARE HINDFAIL	PIC X(1).	1154
DECLARE HCOMMENT1	PIC X(80).	1155
DECLARE HCOMMENT2	PIC X(80).	1156
DECLARE HCOMMENT3	PIC X(80).	1157
DECLARE HDLASTMOD	USAGE DATE	1158
	EDIT STRING X(23).	1159
DECLARE HMODPROC	PIC X(20).	1160
!		1161
!		1162
=====		1163
!		1164
! VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE		1165
!		1166

Datatrieve Procedure FIP_MODIFY (cont.)

```

===== 1167
! 1168
DECLARE CAL      USAGE DATE 1169
          EDIT_STRING X(23). 1170
! 1171
! 1172
===== 1173
! 1174
!   READY THE DOMAINS CONNECTIONS, FAILUREMODES, SYSTEMS, AND 1175
!   PROPAGATIONS_FORM 1176
! 1177
===== 1178
! 1179
SET ABORT 1180
READY CONNECTIONS      SHARED READ 1181
READY FAILUREMODES     SHARED READ 1182
READY SYSTEMS          SHARED WRITE 1183
READY PROPAGATIONS_FORM SHARED READ 1184
! 1185
! 1186
===== 1187
! 1188
!   INITIALIZE THE MESSAGE VARIABLES AND INITIALIZE THE COUNTER (KCNT) 1189
!   USED FOR NUMBERING THE LOG FILE RECORDS 1190
! 1191
===== 1192
! 1193
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 1194
TMSG2 = "DATA MUST BE ENTERED IN AT LEAST ONE OF THE FIELDS OR CONTINUE " | 1195
      "MUST BE N" 1196
TMSG3 = "CODE NUMBER IS NOT VALID -- SYSTEM A OR SYSTEM B MUST BE" 1197
TMSG4 = "CODE NUMBER IS NOT VALID -- NOT IN DOMAIN CONNECTIONS" 1198
TMSG5 = "SIGNAL QUALITY IS NOT VALID -- MUST BE 0, 1, 2, 3, 4 OR 5" 1199
TMSG6 = "FMCODE IS NOT VALID -- SOURCE SYSTEM MUST BE" 1200
TMSG7 = "FMCODE IS NOT VALID -- NOT IN DOMAIN FAILUREMODES" 1201
TMSG8 = "SIGNAL TYPE IS NOT VALID -- NOT IN TABLE SIGNAL_TABLE" 1202
TMSG9 = "DIMENSIONS IS NOT VALID -- MUST BE 1, 2 OR 3" 1203
TMSG10 = "MAX. FREQ/TIME " | 1204
      "MUST BE LEFT-JUSTIFIED NUMERIC IN RANGE -99 TO 99 INCLUSIVE" 1205
TMSG11 = "MIN. FREQ/TIME " | 1206
      "MUST BE LEFT-JUSTIFIED NUMERIC IN RANGE -99 TO 99 INCLUSIVE" 1207
TMSG12 = "PARAMETER IS NOT VALID -- NOT IN PARAMETER_TABLE" 1208
TMSG13 = "SYMPTOM DURATION " | 1209
      "MUST BE LEFT-JUSTIFIED NUMERIC IN RANGE -99 TO 99 INCLUSIVE" 1210
TMSG14 = "PERIOD OF ONSET " | 1211
      "MUST BE LEFT-JUSTIFIED NUMERIC IN RANGE -99 TO 99 INCLUSIVE" 1212
TMSG15 = "INDICATES FAILURE IS NOT VALID -- MUST BE T OR F" 1213
TMSG16 = "CONTINUE IS NOT VALID -- MUST BE Y OR N" 1214
TMSG17 = "NO RECORDS HAVE BEEN FOUND WITH THE DATA INDICATED ABOVE" 1215
TMSG18 = "ENTER MODIFICATIONS IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 1216
TMSG19 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A" 1217

```

Datatrieve Procedure FIP_MODIFY (cont.)

KCNT	= 0	1218
:		1219
:		1220
=====		1221
:		1222
:	THIS SECTION CALLS PROCEDURE FIP_MODIFY_1 TO REQUEST A SYSTEM, READY	1223
:	THE CORRESPONDING DOMAIN AND CALL PROCEDURE FIP_MODIFY_2 TO REQUEST	1224
:	DATA TO IDENTIFY THE FAILURE INFORMATION PROPAGATIONS WHICH ARE TO BE	1225
:	MODIFIED FOR THE DESIGNATED SYSTEM.	1226
:		1227
=====		1228
:		1229
:	FIP_MODIFY_1	1230
:	FN\$DELETE_LOG("PROPAGATIONS")	1231
:	FN\$DELETE_LOG("PROPAGATIONS_FILE")	1232
:	FN\$DELETE_LOG("PROC")	1233
:	END-PROCEDURE	1234

Datatrieve Procedure FIP_MODIFY_1 (cont.)

```

IF FLG1 = "Y" THEN                                     1286
  BEGIN                                               1287
    FOR SYSTEMS WITH SYSTEM = TSYSTEM                1288
      BEGIN                                           1289
        IF PROPAGATIONS_FILE_CREATED NE "YES" THEN  1290
          FLG1 = "N"                                  1291
        END                                           1292
      IF FLG1 = "N" THEN                               1293
        BEGIN                                         1294
          PRINT SKIP 2,                               1295
            "A failure information propagation file has not been", 1296
          SKIP 1,                                     1297
            "created for System", SPACE 1, TSYSTEM (-) USING X(4), 1298
          SPACE 0, ".", SKIP 1                       1299
          TCONTINUE1 = "X"                            1300
          WHILE TCONTINUE1 NE "Y" AND                 1301
            TCONTINUE1 NE "N"                         1302
            BEGIN                                     1303
              PRINT SKIP 1, "Do you wish to continue?", SKIP 1 1304
              TCONTINUE1 = FN$UPCASE(*."Y or N")        1305
              PRINT " "                                1306
            END                                         1307
          IF TCONTINUE1 = "N" THEN                     1308
            BEGIN                                     1309
              CAL = "NOW"                              1310
              PRINT NEW_PAGE                           1311
              :CLRSCRN                                 1312
              PRINT SKIP 3,                             1313
                "  MODIFY PROPAGATIONS", SKIP 2,        1314
                "-----", SKIP 2,                     1315
                "===== END: ", SPACE 0,              1316
                CAL (-) USING X(17),                   1317
                SPACE 0, " =====", SKIP 2,          1318
                "-----", SKIP 3                       1319
            ABORT                                       1320
          END                                           1321
        END                                           1322
      END                                           1323
    END                                           1324
  END                                           1325
!
! ===== 1326
!
! DATATRIEVE LOGICALS ARE CREATED TO REPRESENT THE NAMES OF THE 1327
! PROPAGATIONS DOMAIN AND DATA FILE ASSOCIATED WITH THE CURRENT 1328
! SYSTEM. THE DOMAIN IS NAMED "PROPAGATIONS_X999" WHERE "X999" 1329
! IS THE CURRENT SYSTEM. THE DATA FILE IS NAMED 1330
! "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_X999.DAT" 1331
!
! ===== 1332
!
! 1333
! 1334
! 1335
! 1336

```

```

IF FN$TRANS LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN 1337
  FN$DELETE LOG("PROPAGATIONS") 1338
FN$CREATE LOG("PROPAGATIONS", "PROPAGATIONS " | TSYSTEM) 1339
IF FN$TRANS LOG("PROPAGATIONS FILE") NE "PROPAGATIONS_FILE" THEN 1340
  FN$DELETE LOG("PROPAGATIONS_FILE") 1341
FN$CREATE LOG("PROPAGATIONS_FILE", 1342
  "DEV$206:[BCDSSME2.DATA]PROPAGATIONS_" | TSYSTEM | ".DAT") 1343
! 1344
! 1345
===== 1346
! 1347
  READY THE PROPAGATIONS DOMAIN FOR THE CURRENT SYSTEM 1348
! 1349
===== 1350
! 1351
  READY PROPAGATIONS SHARED WRITE 1352
! 1353
! 1354
===== 1355
! 1356
! THIS SECTION VERIFIES THAT FAILURE INFORMATION PROPAGATIONS EXIST FOR 1357
! THE CURRENT SYSTEM. IF NO RECORDS ARE FOUND, A RESPONSE IS REQUESTED 1358
! TO CONTINUE MODIFYING FAILURE INFORMATION PROPAGATIONS FOR A NEW 1359
! OR TO EXIT PROCEDURE. 1360
! 1361
===== 1362
! 1363
FLG2 = "N" 1364
FOR FIRST 1 PROPAGATIONS 1365
  BEGIN 1366
    FLG2 = "Y" 1367
  END 1368
IF FLG2 = "N" THEN 1369
  BEGIN 1370
    PRINT SKIP 2, 1371
      "Failure information propagations have not been", SKIP 1, 1372
      "entered for System", SPACE 1, 1373
      TSYSTEM (-) USING X(4), SPACE 0, 1374
      ".", SKIP 1 1375
    TCONTINUE2 = "X" 1376
    WHILE TCONTINUE2 NE "Y" AND 1377
      TCONTINUE2 NE "N" 1378
    BEGIN 1379
      PRINT SKIP 1, 1380
        "Do you wish to continue modifying", SKIP 1, 1381
        "FAILURE INFORMATION PROPAGATIONS", SKIP 1, 1382
        "for another system?", SKIP 1 1383
      TCONTINUE2 = FN$UPCASE(*."Y or N") 1384
      PRINT " " 1385
    END 1386
  END 1387
END

```

Datatrieve Procedure FIP_MODIFY_1 (cont.)

```

! 1388
! 1389
!===== 1390
! 1391
! IF FLG2 IS EQUAL TO "Y", PROCEDURE FIP MODIFY 2 IS CALLED TO REQUEST 1392
! INPUT OF DATA TO IDENTIFY THE FAILURE INFORMATION PROPAGATIONS TO BE 1393
! MODIFIED FOR THE CURRENT SYSTEM. IF FLG2 IS "N" AND TCONTINUE2 IS 1394
! "Y", PROCEDURE FIP MODIFY 1 IS CALLED TO REQUEST A NEW SYSTEM. IF 1395
! FLG2 IS "N" AND TCONTINUE2 IS "N", PROCEDURE DTR NULL IS CALLED TO 1396
! TERMINATE MODIFYING FAILURE INFORMATION PROPAGATIONS. 1397
!===== 1398
! 1399
! 1400
IF FN$TRANS_LOG("PROC") NE "PROC" THEN FN$DELETE_LOG("PROC") 1401
IF FLG2 = "Y" THEN 1402
  FN$CREATE_LOG("PROC", "FIP_MODIFY_2") 1403
IF FLG2 = "N" AND 1404
  TCONTINUE2 = "Y" THEN 1405
  FN$CREATE_LOG("PROC", "FIP_MODIFY_1") 1406
IF FLG2 = "N" AND 1407
  TCONTINUE2 = "N" THEN 1408
  FN$CREATE_LOG("PROC", "DTR_NULL") 1409
:PROC 1410
END-PROCEDURE 1411

```

Datatrieve Procedure FIP_MODIFY_2

```

DEFINE PROCEDURE FIP_MODIFY_2
=====
PRIMARY LOOP TO MODIFY PROPAGATIONS
THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:
1. LOOP TO REQUEST SEARCH DATA, TEST VALUES, PROMPT FOR CORRECTION
   OF INVALID INFORMATION AND COUNT THE RECORDS WHICH MATCH THE
   SPECIFIED INPUT FIELDS
2. IF TCONTINUE3 NE "N", SECTION TO DISPLAY THE MATCHING RECORDS
   ONE AT A TIME FOR POSSIBLE MODIFICATION (THIS SECTION IS
   TERMINATED WHEN TCONTINUE4 = "N")
3. IF TCONTINUE3 = "N", SECTION TO REQUEST RESPONSE TO CONTINUE
   MODIFYING DATA FOR CURRENT SYSTEM
4. SECTION TO REQUEST RESPONSE TO CONTINUE PROCEDURE OR EXIT TO
   MENU
=====
TCONTINUE3 = "Y"
WHILE TCONTINUE3 = "Y"
  BEGIN
    =====
    LOOP TO DISPLAY BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON
    ON THE FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF
    INVALID DATA
    =====
    FLG3 = "N"
    MSG = 1
    WHILE FLG3 = "N"
      BEGIN
        IF MSG = 1 THEN TMSG = TMSG1
        IF MSG = 2 THEN TMSG = TMSG2
        IF MSG = 3 THEN TMSG = TMSG3 || " " | TSYSTEM
        IF MSG = 4 THEN TMSG = TMSG4
        IF MSG = 5 THEN TMSG = TMSG5
        IF MSG = 6 THEN TMSG = TMSG6 || " " | TSYSTEM
        IF MSG = 7 THEN TMSG = TMSG7
        IF MSG = 8 THEN TMSG = TMSG8
        IF MSG = 9 THEN TMSG = TMSG9
        IF MSG = 10 THEN TMSG = TMSG10
        IF MSG = 11 THEN TMSG = TMSG11
        IF MSG = 12 THEN TMSG = TMSG12
        IF MSG = 13 THEN TMSG = TMSG13

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

IF IMMSG = 14 THEN TMSG = TMSG14      1463
IF IMMSG = 15 THEN TMSG = TMSG15      1464
IF IMMSG = 16 THEN TMSG = TMSG16      1465
IF IMMSG = 17 THEN TMSG = TMSG17      1466
IF IMMSG = 18 THEN TMSG = TMSG18      1467
                                     1468
                                     1469
=====                             1470
|                                     |
| THIS SECTION DISPLAYS THE FIND FAILURE INFORMATION | 1471
| PROPAGATIONS FORM AND RETRIEVES THE DATA ENTERED ON THE | 1472
| FORM |                                     | 1473
|                                     | 1474
|=====                             | 1475
|                                     | 1476
|                                     | 1477
FOR FIRST 1 PROPAGATIONS_FORM          1478
  BEGIN                                1479
    DISPLAY FORM PROPAGATIONS FIN FORM IN 1480
      DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING 1481
      BEGIN                                1482
        IF IMMSG NE 1 THEN                1483
          BEGIN                            1484
            PUT FORM CODE_NUMBER = TCODENO 1485
            PUT FORM FMCODE = TFMCODE      1486
            PUT FORM SIGNAL_TYPE = TSIG     1487
            PUT FORM DIMENSIONS = TDIM      1488
            PUT FORM QUALITY = TSIGQUAL     1489
            PUT FORM MAX_FREQ_TIME = TMAXFT 1490
            PUT FORM MIN_FREQ_TIME = TMINFT 1491
            PUT FORM PARAM = TPAR           1492
            PUT FORM DURATION = TSYMDUR     1493
            PUT FORM ONSET = TPDONSET       1494
            PUT FORM FAILURE = TINDFAIL     1495
            PUT FORM COMMENT_1 = TCOMMENT1  1496
            PUT FORM COMMENT_2 = TCOMMENT2  1497
            PUT FORM COMMENT_3 = TCOMMENT3  1498
          END                              1499
          PUT FORM CONTINUE = TCONTINUE3    1500
          PUT FORM MESSAGE = TMSG           1501
        END RETRIEVE USING                 1502
        BEGIN                              1503
          TCODENO = GET FORM CODE_NUMBER    1504
          TFMCODE = GET FORM FMCODE         1505
          TSIG = GET FORM SIGNAL_TYPE       1506
          TDIM = GET FORM DIMENSIONS        1507
          TSIGQUAL = GET FORM QUALITY       1508
          TMAXFT = GET FORM MAX_FREQ_TIME   1509
          TMINFT = GET FORM MIN_FREQ_TIME   1510
          TPAR = GET FORM PARAM             1511
          TSYMDUR = GET FORM DURATION       1512
          TPDONSET = GET FORM ONSET         1513

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

TINDFAIL      = GET_FORM FAILURE      1514
TCOMMENT1     = GET_FORM COMMENT_1    1515
TCOMMENT2     = GET_FORM COMMENT_2    1516
TCOMMENT3     = GET_FORM COMMENT_3    1517
TCONTINUE3    = GET_FORM CONTINUE     1518
      END                               1519
    END                               1520
  END                               1521
!                               1522
!                               1523
!                               1524
!                               1525
!                               1526
!                               1527
!                               1528
!                               1529
!                               1530
!                               1531
!                               1532
!                               1533
!                               1534
!                               1535
!                               1536
!                               1537
!                               1538
!                               1539
!                               1540
!                               1541
!                               1542
!                               1543
!                               1544
!                               1545
!                               1546
!                               1547
!                               1548
!                               1549
!                               1550
!                               1551
!                               1552
!                               1553
!                               1554
!                               1555
!                               1556
!                               1557
!                               1558
!                               1559
!                               1560
!                               1561
!                               1562
!                               1563
!                               1564

```

IF TCONTINUE3 IS NOT EQUAL TO "N", ANY LEADING BLANKS
 WHICH WERE INADVERTANTLY ENTERED IN TSIG, TPAR,
 TCOMMENT1, TCOMMENT2 AND TCOMMENT3 ARE REMOVED

```

IF TCONTINUE3 NE "N" THEN
  BEGIN
    I = 1
    WHILE I LE 5
      BEGIN
        IF I = 1 THEN TEMP = TSIG
        IF I = 2 THEN TEMP = TPAR
        IF I = 3 THEN TEMP = TCOMMENT1
        IF I = 4 THEN TEMP = TCOMMENT2
        IF I = 5 THEN TEMP = TCOMMENT3
        IF TEMP NE " " THEN
          BEGIN
            J = 1
            WHILE FN$STR_EXTRACT(TEMP, J, 1) = " "
              BEGIN
                J = J + 1
              END
            TEMP = FN$STR_EXTRACT(TEMP, J, 249 - J + 1)
          END
        IF I = 1 THEN TSIG      = TEMP
        IF I = 2 THEN TPAR      = TEMP
        IF I = 3 THEN TCOMMENT1 = TEMP
        IF I = 4 THEN TCOMMENT2 = TEMP
        IF I = 5 THEN TCOMMENT3 = TEMP
        I = I + 1
      END
    END
  END

```

IF TCONTINUE3 IS NOT EQUAL TO "N", TESTS ARE PERFORMED TO
 VERIFY ANY DATA ENTERED ON THE FORM. AS APPROPRIATE, THE
 PROCEDURE CHECKS ANY OR ALL OF THE FOLLOWING:

Datatrieve Procedure FIP_MODIFY_2 (cont.)

- | | |
|--|------|
| 1. DATA HAS BEEN ENTERED IN AT LEAST ONE OF THE
FIELDS | 1565 |
| 2. TSYSA OR TSYSB IS EQUAL TO TSYSTEM | 1566 |
| 3. TCODENO EXISTS IN DOMAIN CONNECTIONS | 1567 |
| 4. TSIGQUAL IS NUMERIC AND EQUAL TO 0, 1, 2, 3, 4
OR 5 | 1568 |
| 5. TSSYS IS EQUAL TO TSYSTEM | 1569 |
| 6. TFMCODE EXISTS IN DOMAIN FAILUREMODES | 1570 |
| 7. TSIG IS IN TABLE SIGNAL_TABLE | 1571 |
| 8. TDIM IS NUMERIC AND EQUAL TO 1, 2 OR 3 | 1572 |
| 9. TMAXFT IS SIGNED NUMERIC IN RANGE -99 TO 99
INCLUSIVE | 1573 |
| 10. TMINFT IS SIGNED NUMERIC IN RANGE -99 TO 99
INCLUSIVE | 1574 |
| 11. TPAR IS IN TABLE PARAMETER_TABLE | 1575 |
| 12. TSYMDUR IS SIGNED NUMERIC IN RANGE -99 TO 99
INCLUSIVE | 1576 |
| 13. TPDONSET IS SIGNED NUMERIC IN RANGE -99 TO 99
INCLUSIVE | 1577 |
| 14. TINDFAIL IS "T" OR "F" | 1578 |
| 15. TCONTINUE3 IS "Y" OR "N" | 1579 |
| 16. AT LEAST ONE RECORD EXISTS WITH THE DATA SPECIFIED | 1580 |

```

=====
NDIM      = 0
NSIGQUAL  = 0
NMAXFT    = 0
NMINFT    = 0
NSYMDUR   = 0
NPDONSET  = 0
FLG3 = "Y"
IF TCONTINUE3 NE "N" THEN
  BEGIN
    IF TCODENO = " " AND TFMCODE = " " AND
       TSIG = " " AND TDIM = " " AND
       TSIGQUAL = " " AND TMAXFT = " " AND
       TMINFT = " " AND TPAR = " " AND
       TSYMDUR = " " AND TPDONSET = " " AND
       TINDFAIL = " " AND TCOMMENT1 = " " AND
       TCOMMENT2 = " " AND TCOMMENT3 = " " THEN
      BEGIN
        FLG3 = "N"
        IMSG = 2
      END
    IF FLG3 = "Y" AND TCODENO NE " " THEN
      BEGIN
        TSYSA = FN$STR_EXTRACT(TCODENO, 1, 4)
        TSYSB = FN$STR_EXTRACT(TCODENO, 14, 4)
        IF TSYSA NE TSYSTEM AND
           TSYSB NE TSYSTEM THEN

```


Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

        BEGIN                                1616
            FLG3 = "N"                        1617
            MSG = 3                          1618
        END                                  1619
    END                                    1620
IF FLG3 = "Y" AND TCODENO NE " " THEN      1621
    BEGIN                                  1622
        TSYSMODA = FN$STR_EXTRACT(TCODENO, 1, 8) 1623
        TCONNECTU = FN$STR_EXTRACT(TCODENO, 9, 5) 1624
        TSYSMODB = FN$STR_EXTRACT(TCODENO, 14, 8) 1625
        IF TSYSMODA > TSYSMODB THEN              1626
            BEGIN                                1627
                TCODENO = TSYSMODB | TCONNECTU | TSYSMODA 1628
            END                                    1629
            FLG4 = "N"                            1630
            FOR CONNECTIONS WITH CODE_NUMBER = TCODENO      1631
                BEGIN                                1632
                    FLG4 = "Y"                      1633
                END                                    1634
            IF FLG4 = "N" THEN                      1635
                BEGIN                                1636
                    FLG3 = "N"                      1637
                    MSG = 4                          1638
                END                                    1639
            END                                    1640
        END                                    1641
    IF FLG3 = "Y" AND TSIGQUAL NE " " THEN      1642
        BEGIN                                  1643
            FLG4 = "N"                            1644
            IF TSIGQUAL IN NUMBER_TABLE THEN      1645
                BEGIN                                1646
                    NSIGQUAL = TSIGQUAL              1647
                    IF NSIGQUAL GE 0 AND              1648
                        NSIGQUAL LE 5 THEN FLG4 = "Y" 1649
                END                                    1650
            IF FLG4 = "N" THEN                      1651
                BEGIN                                1652
                    FLG3 = "N"                      1653
                    MSG = 5                          1654
                END                                    1655
            END                                    1656
        END                                    1657
    IF FLG3 = "Y" AND TFMCODE NE " " THEN      1658
        BEGIN                                  1659
            TSSYS = FN$STR_EXTRACT(TFMCODE, 1, 4) 1660
            IF TSSYS NE TSYSYSTEM THEN            1661
                BEGIN                                1662
                    FLG3 = "N"                      1663
                    MSG = 6                          1664
                END                                    1665
            END                                    1666
        END                                    1666
    IF FLG3 = "Y" AND TFMCODE NE " " THEN      1666
        BEGIN

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

      FLG4 = "N"                                1667
      FOR FAILUREMODES WITH FMCODE = TFMCODE    1668
      BEGIN                                     1669
          FLG4 = "Y"                            1670
      END                                       1671
      IF FLG4 = "N" THEN                       1672
      BEGIN                                     1673
          FLG3 = "N"                            1674
          IMSG = 7                              1675
      END                                       1676
      END                                       1677
      IF FLG3 = "Y" AND TSIG NE " " THEN      1678
      BEGIN                                     1679
          IF TSIG NOT IN SIGNAL_TABLE THEN    1680
          BEGIN                                 1681
              FLG3 = "N"                      1682
              IMSG = 8                        1683
          END                                   1684
      END                                       1685
      IF FLG3 = "Y" AND TDIM NE " " THEN      1686
      BEGIN                                     1687
          FLG4 = "N"                            1688
          IF TDIM IN NUMBER_TABLE THEN        1689
          BEGIN                                 1690
              NDIM = TDIM                     1691
              IF NDIM GE 1 AND                 1692
              NDIM LE 3 THEN FLG4 = "Y"       1693
          END                                   1694
          IF FLG4 = "N" THEN                   1695
          BEGIN                                 1696
              FLG3 = "N"                      1697
              IMSG = 9                        1698
          END                                   1699
      END                                       1700
      IF FLG3 = "Y" AND TMAXFT NE " " THEN    1701
      BEGIN                                     1702
          FLG4 = "N"                            1703
          TMAXFT1 = FN$STR_EXTRACT(TMAXFT,1,1) 1704
          TMAXFT2 = FN$STR_EXTRACT(TMAXFT,2,1) 1705
          TMAXFT3 = FN$STR_EXTRACT(TMAXFT,3,1) 1706
          IF TMAXFT1 IN SIGN_TABLE AND        1707
          TMAXFT2 IN NUMBER_TABLE AND        1708
          TMAXFT3 IN NUMBER_TABLE THEN      1709
          BEGIN                                 1710
              NMAXFT = TMAXFT                 1711
              FLG4 = "Y"                     1712
          END                                   1713
          IF TMAXFT1 IN SIGN_TABLE AND        1714
          TMAXFT2 IN NUMBER_TABLE AND        1715
          TMAXFT3 = " " THEN                 1716
          BEGIN                                 1717

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

        NMAXFT = TMAXFT                                1718
        FLG4   = "Y"                                    1719
    END                                                  1720
    IF TMAXFT1 IN NUMBER_TABLE AND                      1721
       TMAXFT2 IN NUMBER_TABLE AND                     1722
       TMAXFT3 = " " THEN                               1723
    BEGIN                                              1724
        NMAXFT = TMAXFT                                1725
        FLG4   = "Y"                                    1726
    END                                                  1727
    IF TMAXFT1 IN NUMBER_TABLE AND                      1728
       TMAXFT2 = " " AND                                1729
       TMAXFT3 = " " THEN                               1730
    BEGIN                                              1731
        NMAXFT = TMAXFT                                1732
        FLG4   = "Y"                                    1733
    END                                                  1734
    IF FLG4 = "N" THEN                                  1735
    BEGIN                                              1736
        FLG3 = "N"                                       1737
        IMSG = 10                                        1738
    END                                                  1739
END                                                    1740
IF FLG3 = "Y" AND TMINFT NE " " THEN                  1741
BEGIN                                                  1742
    FLG4 = "N"                                           1743
    TMINFT1 = FN$STR_EXTRACT(TMINFT,1,1)                1744
    TMINFT2 = FN$STR_EXTRACT(TMINFT,2,1)                1745
    TMINFT3 = FN$STR_EXTRACT(TMINFT,3,1)                1746
    IF TMINFT1 IN SIGN_TABLE AND                        1747
       TMINFT2 IN NUMBER_TABLE AND                     1748
       TMINFT3 IN NUMBER_TABLE THEN                    1749
    BEGIN                                              1750
        NMINFT = TMINFT                                1751
        FLG4   = "Y"                                    1752
    END                                                  1753
    IF TMINFT1 IN SIGN_TABLE AND                        1754
       TMINFT2 IN NUMBER_TABLE AND                     1755
       TMINFT3 = " " THEN                               1756
    BEGIN                                              1757
        NMINFT = TMINFT                                1758
        FLG4   = "Y"                                    1759
    END                                                  1760
    IF TMINFT1 IN NUMBER_TABLE AND                      1761
       TMINFT2 IN NUMBER_TABLE AND                     1762
       TMINFT3 = " " THEN                               1763
    BEGIN                                              1764
        NMINFT = TMINFT                                1765
        FLG4   = "Y"                                    1766
    END                                                  1767
    IF TMINFT1 IN NUMBER_TABLE AND                      1768

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

        TMINFT2 = " "          AND          1769
        TMINFT3 = " "          THEN         1770
        BEGIN                      1771
            NMINFT = TMINFT        1772
            FLG4   = "Y"          1773
        END                      1774
    IF FLG4 = "N" THEN            1775
        BEGIN                      1776
            FLG3 = "N"            1777
            IMSG = 11             1778
        END                      1779
    END                          1780
IF FLG3 = "Y" AND TMAXFT NE " " AND TMINFT NE " " THEN 1781
    BEGIN                      1782
        IF NMAXFT < NMINFT THEN 1783
            BEGIN              1784
                HMINFT = TMAXFT 1785
                TMAXFT = TMINFT 1786
                TMINFT = HMINFT 1787
                NMAXFT = TMAXFT 1788
                NMINFT = TMINFT 1789
            END                1790
        END                  1791
    IF FLG3 = "Y" AND TPAR NE " " THEN 1792
        BEGIN                  1793
            IF TPAR NOT IN PARAMETER_TABLE THEN 1794
                BEGIN          1795
                    FLG3 = "N" 1796
                    IMSG = 12   1797
                END            1798
            END              1799
        END                  1800
    IF FLG3 = "Y" AND TSYPDUR NE " " THEN 1801
        BEGIN                  1802
            FLG4 = "N"          1803
            TSYPDUR1 = FN$STR_EXTRACT(TSYMDUR,1,1) 1804
            TSYPDUR2 = FN$STR_EXTRACT(TSYMDUR,2,1) 1805
            TSYPDUR3 = FN$STR_EXTRACT(TSYMDUR,3,1) 1806
            IF TSYPDUR1 IN SIGN_TABLE AND          1807
               TSYPDUR2 IN NUMBER_TABLE AND        1808
               TSYPDUR3 IN NUMBER_TABLE THEN        1809
                BEGIN          1810
                    NSYMDUR = TSYMDUR 1811
                    FLG4   = "Y"      1812
                END            1813
            IF TSYMDUR1 IN SIGN_TABLE AND          1814
               TSYMDUR2 IN NUMBER_TABLE AND        1815
               TSYMDUR3 = " " THEN                  1816
                BEGIN          1817
                    NSYMDUR = TSYMDUR 1818
                    FLG4   = "Y"      1819
                END
            END
        END
    END

```

IF TSYMDUR1 IN NUMBER_TABLE AND	1820
TSYMDUR2 IN NUMBER_TABLE AND	1821
TSYMDUR3 = " " THEN	1822
BEGIN	1823
NSYMDUR = TSYMDUR	1824
FLG4 = "Y"	1825
END	1826
IF TSYMDUR1 IN NUMBER_TABLE AND	1827
TSYMDUR2 = " " AND	1828
TSYMDUR3 = " " THEN	1829
BEGIN	1830
NSYMDUR = TSYMDUR	1831
FLG4 = "Y"	1832
END	1833
IF FLG4 = "N" THEN	1834
BEGIN	1835
FLG3 = "N"	1836
IMSG = 13	1837
END	1838
END	1839
IF FLG3 = "Y" AND TPDONSET NE " " THEN	1840
BEGIN	1841
FLG4 = "N"	1842
TPDONSET1 = FN\$STR_EXTRACT(TPDONSET,1,1)	1843
TPDONSET2 = FN\$STR_EXTRACT(TPDONSET,2,1)	1844
TPDONSET3 = FN\$STR_EXTRACT(TPDONSET,3,1)	1845
IF TPDONSET1 IN SIGN_TABLE AND	1846
TPDONSET2 IN NUMBER_TABLE AND	1847
TPDONSET3 IN NUMBER_TABLE THEN	1848
BEGIN	1849
NPDONSET = TPDONSET	1850
FLG4 = "Y"	1851
END	1852
IF TPDONSET1 IN SIGN_TABLE AND	1853
TPDONSET2 IN NUMBER_TABLE AND	1854
TPDONSET3 = " " THEN	1855
BEGIN	1856
NPDONSET = TPDONSET	1857
FLG4 = "Y"	1858
END	1859
IF TPDONSET1 IN NUMBER_TABLE AND	1860
TPDONSET2 IN NUMBER_TABLE AND	1861
TPDONSET3 = " " THEN	1862
BEGIN	1863
NPDONSET = TPDONSET	1864
FLG4 = "Y"	1865
END	1866
IF TPDONSET1 IN NUMBER_TABLE AND	1867
TPDONSET2 = " " AND	1868
TPDONSET3 = " " THEN	1869
BEGIN	1870

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

        NPDONSET = TPDONSET                                1871
        FLG4     = "Y"                                     1872
    END                                                    1873
    IF FLG4 = "N" THEN                                     1874
    BEGIN                                                  1875
        FLG3 = "N"                                         1876
        IMMSG = 14                                         1877
    END                                                    1878
END                                                        1879
IF FLG3 = "Y" AND TINDFAIL NE " " THEN                    1880
BEGIN                                                    1881
    IF TINDFAIL NE "T" AND                                1882
       TINDFAIL NE "F" THEN                              1883
    BEGIN                                                1884
        FLG3 = "N"                                         1885
        IMMSG = 15                                         1886
    END                                                  1887
END                                                        1888
IF FLG3 = "Y" THEN                                       1889
BEGIN                                                    1890
    IF TCONTINUE3 NE "Y" AND                              1891
       TCONTINUE3 NE "N" THEN                            1892
    BEGIN                                                1893
        FLG3 = "N"                                         1894
        IMMSG = 16                                         1895
    END                                                  1896
END                                                        1897
IF FLG3 = "Y" THEN                                       1898
BEGIN                                                    1899
    FLG4 = "N"                                             1900
    IF FLG4 = "N" AND                                     1901
       TFMCODE NE " " AND                                1902
       TCODENO NE " " AND                                1903
       TSIG     NE " " THEN                              1904
    BEGIN                                                1905
        JCNT = 0                                           1906
        FOR PROPAGATIONS WITH FMCODE                     1907
                                = TFMCODE AND              1908
                                CODE_NUMBER = TCODENO AND  1909
                                SIGNAL_TYPE = TSIG        1910
        BEGIN                                            1911
            FLG5 = "Y"                                     1912
            :FIP_MODIFY_3                                1912
            IF FLG5 = "Y" THEN                            1913
                JCNT = JCNT + 1                          1914
            END                                           1915
            FLG4 = "Y"                                     1916
        END                                              1917
    IF FLG4 = "N" AND                                     1918
       TFMCODE NE " " AND                                1919
       TCODENO NE " " THEN                              1920
    BEGIN                                                1921

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

JCNT = 0	1922
FOR PROPAGATIONS WITH FMCODE = TFMCODE AND	1923
CODE_NUMBER = TCODENO	1924
BEGIN	1925
FLG5 = "Y"	1926
:FIP_MODIFY_3	1927
IF FLG5 = "Y" THEN	1928
JCNT = JCNT + 1	1929
END	1930
FLG4 = "Y"	1931
END	1932
IF FLG4 = "N" AND	1933
TFMCODE NE " " AND	1934
TSIG NE " " THEN	1935
BEGIN	1936
JCNT = 0	1937
FOR PROPAGATIONS WITH FMCODE = TFMCODE AND	1938
SIGNAL_TYPE = TSIG	1939
BEGIN	1940
FLG5 = "Y"	1941
:FIP_MODIFY_3	1942
IF FLG5 = "Y" THEN	1943
JCNT = JCNT + 1	1944
END	1945
FLG4 = "Y"	1946
END	1947
IF FLG4 = "N" AND	1948
TCODENO NE " " AND	1949
TSIG NE " " THEN	1950
BEGIN	1951
JCNT = 0	1952
FOR PROPAGATIONS WITH CODE_NUMBER = TCODENO AND	1953
SIGNAL_TYPE = TSIG	1954
BEGIN	1955
FLG5 = "Y"	1956
:FIP_MODIFY_3	1957
IF FLG5 = "Y" THEN	1958
JCNT = JCNT + 1	1959
END	1960
FLG4 = "Y"	1961
END	1962
IF FLG4 = "N" AND	1963
TFMCODE NE " " THEN	1964
BEGIN	1965
JCNT = 0	1966
FOR PROPAGATIONS WITH FMCODE = TFMCODE	1967
BEGIN	1968
FLG5 = "Y"	1969
:FIP_MODIFY_3	1970
IF FLG5 = "Y" THEN	1971
JCNT = JCNT + 1	1972

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

                                END                                1973
                                FLG4 = "Y"                        1974
                                END                                1975
                                IF FLG4 = "N" AND                1976
                                TCODENO NE " " THEN              1977
                                BEGIN                              1978
                                JCNT = 0                          1979
                                FOR PROPAGATIONS WITH CODE_NUMBER = TCODENO 1980
                                BEGIN                              1981
                                FLG5 = "Y"                        1982
                                :FIP_MODIFY_3                    1983
                                IF FLG5 = "Y" THEN                1984
                                JCNT = JCNT + 1                  1985
                                END                                1986
                                FLG4 = "Y"                        1987
                                END                                1988
                                IF FLG4 = "N" AND                1989
                                TSIG NE " "                      1990
                                BEGIN                              1991
                                JCNT = 0                          1992
                                FOR PROPAGATIONS WITH SIGNAL_TYPE = TSIG 1993
                                BEGIN                              1994
                                FLG5 = "Y"                        1995
                                :FIP_MODIFY_3                    1996
                                IF FLG5 = "Y" THEN                1997
                                JCNT = JCNT + 1                  1998
                                END                                1999
                                FLG4 = "Y"                        2000
                                END                                2001
                                IF FLG4 = "N"                    2002
                                BEGIN                              2003
                                JCNT = 0                          2004
                                FOR PROPAGATIONS                2005
                                BEGIN                              2006
                                FLG5 = "Y"                        2007
                                :FIP_MODIFY_3                    2008
                                IF FLG5 = "Y" THEN                2009
                                JCNT = JCNT + 1                  2010
                                END                                2011
                                END                                2012
                                IF JCNT = 0 THEN                  2013
                                BEGIN                              2014
                                FLG3 = "N"                        2015
                                IMSG = 17                         2016
                                END                                2017
                                END                                2018
                                END                                2019
                                END                                2020
                                !                                  2021
                                !                                  2022
                                !                                  2023
=====

```


IF TCONTINUE3 NE "N"

PARAMETER,

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

                                DATE_CREATED      2075
        BEGIN                                2076
            IF TCONTINUE4 = "Y" THEN            2077
                BEGIN                            2078
                    FLG5 = "Y"                    2079
                    :FIP_MODIFY_3                2080
                    IF FLG5 = "Y" THEN            2081
                        BEGIN                    2082
                            ICNT = ICNT + 1      2083
                            :FIP_MODIFY_4        2084
                        END                        2085
                    END                            2086
                END                                2087
                FLG4 = "Y"                        2088
            END                                    2089
        IF FLG4 = "N" AND                        2090
           TFMCODE NE " " AND                    2091
           TSIG NE " " THEN                      2092
            BEGIN                                2093
                ICNT = 0                          2094
                FOR PROPAGATIONS WITH FMCODE      2095
                    = TFMCODE AND                2096
                    SIGNAL_TYPE = TSIG           2097
                    SORTED BY                    2098
                    FMCODE,                      2099
                    SIGNAL_TYPE,                 2100
                    PARAMETER,                   2101
                    DATE_CREATED                 2102
                BEGIN                            2103
                    IF TCONTINUE4 = "Y" THEN      2104
                        BEGIN                    2105
                            FLG5 = "Y"            2106
                            :FIP_MODIFY_3        2107
                            IF FLG5 = "Y" THEN    2108
                                BEGIN            2109
                                    ICNT = ICNT + 1  2110
                                    :FIP_MODIFY_4    2111
                                END                2112
                            END                    2113
                        END                            2114
                    END                                2115
                END                                    2116
                FLG4 = "Y"                        2117
            END                                    2118
        IF FLG4 = "N" AND                        2119
           TCODENO NE " " AND                    2120
           TSIG NE " " THEN                      2121
            BEGIN                                2122
                ICNT = 0                          2123
                FOR PROPAGATIONS WITH CODE_NUMBER = TCODENO AND 2124
                    SIGNAL_TYPE = TSIG           2125
                    SORTED BY                    2126
                    FMCODE,                      2127
                    SIGNAL_TYPE,                 2128
                    PARAMETER,                   2129
                    DATE_CREATED                 2130
            END

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

```

BEGIN                                                    2126
  IF TCONTINUE4 = "Y" THEN                              2127
    BEGIN                                                2128
      FLG5 = "Y"                                         2129
      :FIP_MODIFY_3                                     2130
      IF FLG5 = "Y" THEN                                2131
        BEGIN                                           2132
          ICNT = ICNT + 1                                2133
          :FIP_MODIFY_4                                  2134
        END                                             2135
      END                                               2136
    END                                                 2137
    FLG4 = "Y"                                           2138
  END                                                  2139
  IF FLG4 = "N" AND                                     2140
    TFMCODE NE " " THEN                                2141
    BEGIN                                              2142
      ICNT = 0                                           2143
      FOR PROPAGATIONS WITH FMCODE = TFMCODE SORTED BY 2144
        FMCODE,                                         2145
        SIGNAL_TYPE,                                   2146
        PARAMETER,                                     2147
        DATE_CREATED                                  2148
      BEGIN                                           2149
        IF TCONTINUE4 = "Y" THEN                      2150
          BEGIN                                        2151
            FLG5 = "Y"                                2152
            :FIP_MODIFY_3                              2153
            IF FLG5 = "Y" THEN                        2154
              BEGIN                                    2155
                ICNT = ICNT + 1                        2156
                :FIP_MODIFY_4                          2157
              END                                      2158
            END                                        2159
          END                                          2160
        END                                          2161
        FLG4 = "Y"                                    2162
      END                                          2163
      IF FLG4 = "N" AND                                2164
        TCODENO NE " " THEN                          2165
        BEGIN                                          2166
          ICNT = 0                                     2166
          FOR PROPAGATIONS WITH CODE_NUMBER = TCODENO SORTED BY 2167
            FMCODE,                                     2168
            SIGNAL_TYPE,                               2169
            PARAMETER,                                 2170
            DATE_CREATED                              2171
          BEGIN                                       2172
            IF TCONTINUE4 = "Y" THEN                 2173
              BEGIN                                   2174
                FLG5 = "Y"                             2175
                :FIP_MODIFY_3                         2176
              END
            END
          END
        END
      END
    END
  END

```



```

END
END
END
=====
IF TCONTINUE3 EQUALS "N", A RESPONSE IS REQUESTED TO CONTINUE
MODIFYING FAILURE INFORMATION PROPAGATIONS FOR SYSTEM = TSYSTEM
=====
IF TCONTINUE3 = "N" THEN
  BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    TCONTINUE3 = "X"
    WHILE TCONTINUE3 NE "Y" AND
      TCONTINUE3 NE "N"
      BEGIN
        PRINT SKIP 2,
          "Do you wish to continue modifying", SKIP 1,
          "FAILURE INFORMATION PROPAGATIONS", SKIP 1,
          "for System", SPACE 1, TSYSTEM (-) USING X(4),
          SPACE 0, "?", SKIP 1
        TCONTINUE3 = FN$UPCASE(*."Y or N")
        PRINT " "
      END
    END
  END
END
=====
SECTION TO REQUEST RESPONSE TO CONTINUE MODIFYING FAILURE INFORMATION
PROPAGATIONS FOR A NEW SYSTEM OR TO EXIT PROCEDURE. IF TCONTINUE5 IS
EQUAL TO "Y", PROCEDURE FIP_MODIFY_1 IS CALLED TO REQUEST A NEW
SYSTEM. IF TCONTINUE5 IS EQUAL TO "N", PROCEDURE DTR_NULL IS CALLED
TO TERMINATE MODIFYING FAILURE INFORMATION PROPAGATIONS.
=====
TCONTINUE5 = "X"
WHILE TCONTINUE5 NE "Y" AND
  TCONTINUE5 NE "N"
  BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    PRINT SKIP 3,
      "Do you wish to continue modifying", SKIP 1,
      "FAILURE INFORMATION PROPAGATIONS", SKIP 1,

```

Datatrieve Procedure FIP_MODIFY_2 (cont.)

"for another system?", SKIP 1	2279
TCONTINUE5 = FN\$UPCASE(*."Y or N")	2280
PRINT " "	2281
END	2282
IF FN\$TRANS_LOG("PROC") NE "PROC" THEN FN\$DELETE_LOG("PROC")	2283
CHOICE	2284
TCONTINUE5 = "Y" THEN FN\$CREATE_LOG("PROC", "FIP_MODIFY_1")	2285
TCONTINUE5 = "N" THEN FN\$CREATE_LOG("PROC", "DTR_NULL")	2286
END_CHOICE	2287
:PROC	2288
END-PROCEDURE	2289

Datatrieve Procedure FIP_MODIFY_3

```

DEFINE PROCEDURE FIP_MODIFY_3
!
!
!=====
!
! THIS SECTION DETERMINES IF A GIVEN RECORD MATCHES ALL OF THE FIELDS
! ENTERED ON THE FIND FAILURE INFORMATION PROPAGATIONS FORM.  THE VALUES
! OF THE KEY FIELDS (CODE_NUMBER, FMCODE AND SIGNAL_TYPE) ARE NOT
! CHECKED IN THIS SECTION SINCE THESE VALUES HAVE ALREADY BEEN MATCHED
! IN THE "FOR ..." STATEMENT WHICH ESTABLISHES THE RECORD STREAM.
!=====
!
IF TDIM NE " " THEN
  BEGIN
    IF DIMENSIONS NE NDIM THEN FLG5 = "N"
  END
IF TSIGQUAL NE " " THEN
  BEGIN
    IF SIGNAL_QUALITY NE NSIGQUAL THEN FLG5 = "N"
  END
IF TMAXFT NE " " THEN
  BEGIN
    IF MAX_FREQ_OR_TIME NE NMAXFT THEN FLG5 = "N"
  END
IF TMINFT NE " " THEN
  BEGIN
    IF MIN_FREQ_OR_TIME NE NMINFT THEN FLG5 = "N"
  END
IF TPAR NE " " THEN
  BEGIN
    IF PARAMETER NE TPAR THEN FLG5 = "N"
  END
IF TSYMDUR NE " " THEN
  BEGIN
    IF SYMPTOM_DURATION NE NSYMDUR THEN FLG5 = "N"
  END
IF TPDONSET NE " " THEN
  BEGIN
    IF PERIOD_OF_ONSET NE NPDONSET THEN FLG5 = "N"
  END
IF TINDFAIL NE " " THEN
  BEGIN
    IF INDICATES_FAILURE NE TINDFAIL THEN FLG5 = "N"
  END
IF TCOMMENT1 NE " " THEN
  BEGIN
    IF COMMENT1 NE TCOMMENT1 THEN FLG5 = "N"
  END
IF TCOMMENT2 NE " " THEN
  BEGIN

```

Datatrieve Procedure FIP_MODIFY_3 (cont.)

IF COMMENT2 NE TCOMMENT2 THEN FLG5 = "N"	2341
END	2342
IF TCOMMENT3 NE " " THEN	2343
BEGIN	2344
IF COMMENT3 NE TCOMMENT3 THEN FLG5 = "N"	2345
END	2346
END-PROCEDURE	2347

Datatrieve Procedure FIP_MODIFY_4

```

DEFINE PROCEDURE FIP_MODIFY_4                                2348
!                                                            2349
!                                                            2350
!=====                                                    2351
! THE FIELDS OF THE INCOMING FAILURE INFORMATION PROPAGATION RECORD ARE 2352
! ASSIGNED TO VARIABLES FOR DISPLAY AND MODIFICATION. THE FIELDS      2353
! FMCODE, CODE NUMBER, SIGNAL TYPE AND PARAMETER ARE NOT ASSIGNED TO  2354
! VARIABLES SINCE THESE FIELDS CANNOT BE MODIFIED.                  2355
!=====                                                    2356
!                                                            2357
!=====                                                    2358
!                                                            2359
TTDIM      = DIMENSIONS                                     2360
TTSIGQUAL  = SIGNAL QUALITY                                2361
TTMAXFT    = MAX_FREQ_OR_TIME                             2362
TTMINFT    = MIN_FREQ_OR_TIME                             2363
TTSYMDUR   = SYMPTOM_DURATION                             2364
TTPDONSET  = PERIOD_OF_ONSET                               2365
TTINDFAIL  = INDICATES_FAILURE                            2366
TTCOMMENT1 = COMMENT1                                     2367
TTCOMMENT2 = COMMENT2                                     2368
TTCOMMENT3 = COMMENT3                                     2369
!                                                            2370
!                                                            2371
!=====                                                    2372
! LOOP TO DISPLAY A FAILURE INFORMATION PROPAGATION RECORD USING A TDMS 2373
! FORM, RETRIEVE DATA FROM THE FORM, TEST THE INCOMING INFORMATION AND 2374
! REQUEST CORRECTION OF INVALID DATA                                2375
!=====                                                    2376
!                                                            2377
!=====                                                    2378
!                                                            2379
FLG6       = "N"                                           2380
IMSG       = 18                                           2381
WHILE FLG6 = "N"                                           2382
  BEGIN                                                    2383
    IF IMSG = 5 THEN TMSG = TMSG5                          2384
    IF IMSG = 9 THEN TMSG = TMSG9                          2385
    IF IMSG = 15 THEN TMSG = TMSG15                        2386
    IF IMSG = 18 THEN TMSG = TMSG18                       2387
    IF IMSG = 19 THEN TMSG = TMSG19                       2388
!                                                            2389
!=====                                                    2390
! THIS SECTION DISPLAYS THE MODIFY FAILURE INFORMATION              2391
! PROPAGATIONS FORM AND RETRIEVES DATA ENTERED ON THE FORM        2392
!=====                                                    2393
! THIS SECTION DISPLAYS THE MODIFY FAILURE INFORMATION              2394
! PROPAGATIONS FORM AND RETRIEVES DATA ENTERED ON THE FORM        2395
!=====                                                    2396
!                                                            2397
DISPLAY_FORM PROPAGATIONS_MOD_FORM IN                      2398

```


Datatrieve Procedure FIP_MODIFY_4 (cont.)

```

===== 2450
!
! 2451
IF TCONTINUE4 NE "A" THEN 2452
  BEGIN 2453
    I = 1 2454
    WHILE I LE 3 2455
      BEGIN 2456
        IF I = 1 THEN TEMP = TTCOMMENT1 2457
        IF I = 2 THEN TEMP = TTCOMMENT2 2458
        IF I = 3 THEN TEMP = TTCOMMENT3 2459
        IF TEMP NE " " THEN 2460
          BEGIN 2461
            J = 1 2462
            WHILE FN$STR_EXTRACT(TEMP, J, 1) = " " 2463
              BEGIN 2464
                J = J + 1 2465
              END 2466
            TEMP = FN$STR_EXTRACT(TEMP, J, 249 - J + 1) 2467
          END 2468
        IF I = 1 THEN TTCOMMENT1 = TEMP 2469
        IF I = 2 THEN TTCOMMENT2 = TEMP 2470
        IF I = 3 THEN TTCOMMENT3 = TEMP 2471
        I = I + 1 2472
      END 2473
    END 2474
  END 2475
!
! 2476
! 2477
! 2478
! 2479
! 2480
! 2481
! 2482
! 2483
! 2484
! 2485
! 2486
! 2487
! 2488
! 2489
! 2490
! 2491
! 2492
! 2493
! 2494
! 2495
! 2496
! 2497
! 2498
! 2499
! 2500
=====
IF TCONTINUE4 IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
VERIFY THE FOLLOWING:
1. TTSIGQUAL IS EQUAL TO 0, 1, 2, 3, 4 OR 5
2. TTDIM IS EQUAL TO 1, 2 OR 3
3. TTINDFAIL IS EQUAL TO "T" OR "F"
4. TCONTINUE4 IS "Y", "N" OR "A"
=====
FLG6 = "Y"
IF TCONTINUE4 NE "A" THEN
  BEGIN
    IF TTSIGQUAL LT 0 OR
      TTSIGQUAL GT 5 THEN
      BEGIN
        FLG6 = "N"
        IMSG = 5
      END
    IF FLG6 = "Y" THEN
      BEGIN
        IF TTDIM LT 1 OR
          TTDIM GT 3 THEN

```


Datatrieve Procedure FIP_MODIFY_4 (cont.)

```

                                TCOMMENT = FN$STR_EXTRACT(TEMP, 1, J - 1)      2552
                                TEMP      = FN$STR_EXTRACT(TEMP, J, 249 - J + 1) 2553
                                END                                              2554
                                END                                              2555
                                IF I = 1 THEN TTCOMMENT1 = TCOMMENT              2556
                                IF I = 2 THEN TTCOMMENT2 = TCOMMENT              2557
                                IF I = 3 THEN TTCOMMENT3 = TCOMMENT              2558
                                I = I + 1                                       2559
                                END                                              2560
                                END                                              2561
                                !                                              2562
                                !                                              2563
                                ! ===== 2564
                                ! IF TCONTINUE4 IS NOT EQUAL TO "A" AND NEW DATA HAS BEEN ENTERED IN AT 2565
                                ! LEAST ONE OF THE RECORD FIELDS, THIS SECTION PRINTS THE INITIAL RECORD 2566
                                ! DATA TO THE LOG FILE, MODIFIES THE RECORD AND PRINTS THE MODIFIED 2567
                                ! RECORD DATA TO THE LOG FILE (HIGHLIGHTING THE CHANGED FIELDS) 2568
                                ! ===== 2569
                                !                                              2570
                                ! ===== 2571
                                ! IF TCONTINUE4 NE "A" AND 2572
                                ! (DIMENSIONS      NE TTDIM      OR 2573
                                ! SIGNAL_QUALITY.   NE TTSIGQUAL OR 2574
                                ! MAX_FREQ_OR_TIME NE TTMAXFT   OR 2575
                                ! MIN_FREQ_OR_TIME NE TTMINFT   OR 2576
                                ! SYMPTOM_DURATION NE TTSYMDUR  OR 2577
                                ! PERIOD_OF_ONSET  NE TTPDONSET  OR 2578
                                ! INDICATES_FAILURE NE TTINDFAIL OR 2579
                                ! COMMENT1        NE TTCOMMENT1 OR 2580
                                ! COMMENT2        NE TTCOMMENT2 OR 2581
                                ! COMMENT3        NE TTCOMMENT3) THEN 2582
                                ! BEGIN 2583
                                !                                              2584
                                ! ===== 2585
                                ! THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG 2586
                                ! FILE 2587
                                ! ===== 2588
                                !                                              2589
                                ! ===== 2590
                                !                                              2591
                                ! ===== 2592
                                !                                              2593
                                ! ===== 2594
                                ! KCNT = KCNT + 1 2595
                                ! PRINT NEW PAGE, COL 1, "RECORD NO. ", SPACE 0, 2596
                                ! KCNT (-) USING 9(4), SKIP 2, 2597
                                ! "===== ", SPACE 0, 2598
                                ! "===== ", SKIP 1, 2599
                                ! "===== ", SPACE 0, 2600
                                ! "===== ", SKIP 2, 2601
                                ! COL 6, "DATE_CREATED      :", SPACE 1, 2602
                                ! DATE_CREATED (-) USING X(23), SKIP 1,

```

Datatrieve Procedure FIP_MODIFY_4 (cont.)

```

COL 6, "FMCODE          :", SPACE 1,          2603
      FMCODE (-) USING X(20), SKIP 1,          2604
COL 6, "CODE_NUMBER     :", SPACE 1,          2605
      CODE_NUMBER (-) USING X(21), SKIP 1,      2606
COL 6, "SIGNAL_TYPE     :", SPACE 1,          2607
      SIGNAL_TYPE (-) USING X(20), SKIP 1,      2608
COL 6, "SIGNAL_UNITS    :", SPACE 1,          2609
      SIGNAL_UNITS (-) USING X(25), SKIP 1,     2610
COL 6, "DIMENSIONS      :", SPACE 1,          2611
      DIMENSIONS (-) USING 9(1), SKIP 1,        2612
COL 6, "SIGNAL_QUALITY  :", SPACE 1,          2613
      SIGNAL_QUALITY (-) USING 9(1), SKIP 1,    2614
COL 6, "MAX_FREQ_OR_TIME :", SPACE 1,          2615
      MAX_FREQ_OR_TIME (-) USING -Z9, SKIP 1,   2616
COL 6, "MIN_FREQ_OR_TIME :", SPACE 1,          2617
      MIN_FREQ_OR_TIME (-) USING -Z9, SKIP 1,   2618
COL 6, "FT_UNITS        :", SPACE 1,          2619
      FT_UNITS (-) USING X(25), SKIP 1,         2620
COL 6, "PARAMETER       :", SPACE 1,          2621
      PARAMETER (-) USING X(20), SKIP 1,        2622
COL 6, "PARAMETER_UNITS :", SPACE 1,          2623
      PARAMETER_UNITS (-) USING X(25), SKIP 1,  2624
COL 6, "SYMPTOM_DURATION :", SPACE 1,          2625
      SYMPTOM_DURATION (-) USING -Z9, SKIP 1,   2626
COL 6, "PERIOD_OF_ONSET  :", SPACE 1,          2627
      PERIOD_OF_ONSET (-) USING -Z9, SKIP 1,    2628
COL 6, "INDICATES_FAILURE :", SPACE 1,         2629
      INDICATES_FAILURE (-) USING X(1), SKIP 1,  2630
COL 6, "COMMENT1        :", SPACE 1,          2631
      COMMENT1 (-) USING T(50), SKIP 1,         2632
COL 6, "COMMENT2        :", SPACE 1,          2633
      COMMENT2 (-) USING T(50), SKIP 1,         2634
COL 6, "COMMENT3        :", SPACE 1,          2635
      COMMENT3 (-) USING T(50), SKIP 1,         2636
COL 6, "DATE_LAST_MODIFIED :", SPACE 1,        2637
      DATE_LAST_MODIFIED (-) USING X(23), SKIP 1, 2638
COL 6, "MODIFYING_PROCEDURE :", SPACE 1,       2639
      MODIFYING_PROCEDURE (-) USING X(20)       2640

```

```

=====
THIS SECTION STORES THE INITIAL RECORD VALUES FOR USE IN
HIGHLIGHTING THE FIELDS WHICH HAVE BEEN MODIFIED
=====

```

```

HDCREATED = DATE_CREATED
HFMCODE   = FMCODE
HCODENO   = CODE_NUMBER
HSIG      = SIGNAL_TYPE

```

```

2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653

```

Datatrieve Procedure FIP MODIFY 4 (cont.)

HSIGUNIT	=	SIGNAL_UNITS	2654
HDIM	=	DIMENSIONS	2655
HSIGQUAL	=	SIGNAL_QUALITY	2656
HMAXFT	=	MAX_FREQ_OR_TIME	2657
HMINFT	=	MIN_FREQ_OR_TIME	2658
HFTUNIT	=	FT_UNITS	2659
HPAR	=	PARAMETER	2660
HPARUNIT	=	PARAMETER_UNITS	2661
HSYMDUR	=	SYMPTOM_DURATION	2662
HONSET	=	PERIOD_OF_ONSET	2663
HINDFAIL	=	INDICATES_FAILURE	2664
HCOMMENT1	=	COMMENT1	2665
HCOMMENT2	=	COMMENT2	2666
HCOMMENT3	=	COMMENT3	2667
HDLASTMOD	=	DATE_LAST_MODIFIED	2668
HMODPROC	=	MODIFYING_PROCEDURE	2669

THIS SECTION MODIFIES THE FAILURE INFORMATION PROPAGATION USING THE VERIFIED DATA ENTERED ON THE MODIFY FAILURE INFORMATION PROPAGATIONS FORM

```

CAL = "NOW"
MODIFY USING
  BEGIN
    DIMENSIONS = TTDIM
    SIGNAL_QUALITY = TTSIGQUAL
    MAX_FREQ_OR_TIME = TTMAXFT
    MIN_FREQ_OR_TIME = TTMINFT
    SYMPTOM_DURATION = TTSYMDUR
    PERIOD_OF_ONSET = TTPDONSET
    INDICATES_FAILURE = TTINDFAIL
    COMMENT1 = TTCOMMENT1
    COMMENT2 = TTCOMMENT2
    COMMENT3 = TTCOMMENT3
    DATE_LAST_MODIFIED = CAL
    MODIFYING_PROCEDURE = "FIP_MODIFY"
  END

```

THIS SECTION PRINTS THE MODIFIED RECORD DATA TO THE SESSION LOG
FILE AND HIGHLIGHTS THE FIELDS WHICH CONTAIN NEW INFORMATION

Datatrieve Procedure FIP_MODIFY_4 (cont.)

```

PRINT SKIP 1, 2705
"=====", SPACE 0, 2706
"=====", SKIP 2, 2707
COL 1, CHOICE 2708
    DATE_CREATED = HDCREATED THEN " " 2709
    ELSE "****" 2710
END CHOICE, 2711
SPACE 2, "DATE_CREATED :", SPACE 1, 2712
DATE_CREATED (-) USING X(23), SKIP 1, 2713
COL 1, CHOICE 2714
    FMCODE = HFMCODE THEN " " 2715
    ELSE "****" 2716
END CHOICE, 2717
SPACE 2, "FMCODE :", SPACE 1, 2718
FMCODE (-) USING X(20), SKIP 1, 2719
COL 1, CHOICE 2720
    CODE_NUMBER = HCODENO THEN " " 2721
    ELSE "****" 2722
END CHOICE, 2723
SPACE 2, "CODE_NUMBER :", SPACE 1, 2724
CODE_NUMBER (-) USING X(21), SKIP 1, 2725
COL 1, CHOICE 2726
    SIGNAL_TYPE = HSIG THEN " " 2727
    ELSE "****" 2728
END CHOICE, 2729
SPACE 2, "SIGNAL_TYPE :", SPACE 1, 2730
SIGNAL_TYPE (-) USING X(20), SKIP 1, 2731
COL 1, CHOICE 2732
    SIGNAL_UNITS = HSIGUNIT THEN " " 2733
    ELSE "****" 2734
END CHOICE, 2735
SPACE 2, "SIGNAL_UNITS :", SPACE 1, 2736
SIGNAL_UNITS (-) USING X(25), SKIP 1, 2737
COL 1, CHOICE 2738
    DIMENSIONS = HDIM THEN " " 2739
    ELSE "****" 2740
END CHOICE, 2741
SPACE 2, "DIMENSIONS :", SPACE 1, 2742
DIMENSIONS (-) USING 9(1), SKIP 1, 2743
COL 1, CHOICE 2744
    SIGNAL_QUALITY = HSIGQUAL THEN " " 2745
    ELSE "****" 2746
END CHOICE, 2747
SPACE 2, "SIGNAL_QUALITY :", SPACE 1, 2748
SIGNAL_QUALITY (-) USING 9(1), SKIP 1, 2749
COL 1, CHOICE 2750
    MAX_FREQ_OR_TIME = HMAXFT THEN " " 2751
    ELSE "****" 2752
END CHOICE, 2753
SPACE 2, "MAX_FREQ_OR_TIME :", SPACE 1, 2754
MAX_FREQ_OR_TIME (-) USING -Z9, SKIP 1, 2755

```


Datatrieve Procedure FIP_MODIFY_4 (cont.)

COL 1, CHOICE	2756
MIN_FREQ_OR_TIME = HMINFT THEN " "	2757
ELSE "****"	2758
END CHOICE,	2759
SPACE 2, "MIN_FREQ_OR_TIME :", SPACE 1,	2760
MIN_FREQ_OR_TIME (-) USING -Z9, SKIP 1,	2761
COL 1, CHOICE	2762
FT_UNITS = HFTUNIT THEN " "	2763
ELSE "****"	2764
END CHOICE,	2765
SPACE 2, "FT_UNITS :", SPACE 1,	2766
FT_UNITS (-) USING X(25), SKIP 1,	2767
COL 1, CHOICE	2768
PARAMETER = HPAR THEN " "	2769
ELSE "****"	2770
END CHOICE,	2771
SPACE 2, "PARAMETER :", SPACE 1,	2772
PARAMETER (-) USING X(20), SKIP 1,	2773
COL 1, CHOICE	2774
PARAMETER_UNITS = HPARUNIT THEN " "	2775
ELSE "****"	2776
END CHOICE,	2777
SPACE 2, "PARAMETER_UNITS :", SPACE 1,	2778
PARAMETER_UNITS (-) USING X(25), SKIP 1,	2779
COL 1, CHOICE	2780
SYMPTOM_DURATION = HSYMDUR THEN " "	2781
ELSE "****"	2782
END CHOICE,	2783
SPACE 2, "SYMPTOM_DURATION :", SPACE 1,	2784
SYMPTOM_DURATION (-) USING -Z9, SKIP 1,	2785
COL 1, CHOICE	2786
PERIOD_OF_ONSET = HONSET THEN " "	2787
ELSE "****"	2788
END CHOICE,	2789
SPACE 2, "PERIOD_OF_ONSET :", SPACE 1,	2790
PERIOD_OF_ONSET (-) USING -Z9, SKIP 1,	2791
COL 1, CHOICE	2792
INDICATES_FAILURE = HINDFAIL THEN " "	2793
ELSE "****"	2794
END CHOICE,	2795
SPACE 2, "INDICATES_FAILURE :", SPACE 1,	2796
INDICATES_FAILURE (-) USING X(1), SKIP 1,	2797
COL 1, CHOICE	2798
COMMENT1 = HCOMMENT1 THEN " "	2799
ELSE "****"	2800
END CHOICE,	2801
SPACE 2, "COMMENT1 :", SPACE 1,	2802
COMMENT1 (-) USING T(50), SKIP 1,	2803
COL 1, CHOICE	2804
COMMENT2 = HCOMMENT2 THEN " "	2805
ELSE "****"	2806

Datatrieve Procedure FIP_MODIFY_4 (cont.)

```

                END CHOICE,
                SPACE 2, "COMMENT2          :", SPACE 1,
                COMMENT2 (-) USING T(50), SKIP 1,
COL 1, CHOICE
                COMMENT3 = HCOMMENT3 THEN "  "
                ELSE "***"
                END CHOICE,
                SPACE 2, "COMMENT3          :", SPACE 1,
                COMMENT3 (-) USING T(50), SKIP 1,
COL 1, CHOICE
                DATE_LAST_MODIFIED = HDLASTMOD THEN "  "
                ELSE "***"
                END CHOICE,
                SPACE 2, "DATE_LAST_MODIFIED :", SPACE 1,
                DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,
COL 1, CHOICE
                MODIFYING_PROCEDURE = HMODPROC THEN "  "
                ELSE "***"
                END CHOICE,
                SPACE 2, "MODIFYING_PROCEDURE :", SPACE 1,
                MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,
"===== ", SPACE 0,
"===== ", SKIP 1,
"===== ", SPACE 0,
"===== "
: BELL
END
!
!-----
! IF TCONTINUE4 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN MODIFIED. THIS
! MESSAGE WILL APPEAR IN THE SESSION LOG FILE.
!-----
!
IF TCONTINUE4 = "A" THEN
    BEGIN
        PRINT NEW_PAGE, SKIP 3,
        COL 1, "===== ", SPACE 0,
        "===== ", SKIP 10,
        COL 9, "*****", SKIP 2,
        COL 9, "***** RECORD NOT MODIFIED *****", SKIP 2,
        COL 9, "*****", SKIP 10,
        COL 1, "===== ", SPACE 0,
        "===== "
    END
END-PROCEDURE

```

Datatrieve Procedure FIP_STORE

DEFINE PROCEDURE FIP_STORE	2856
!	2857
!	2858
=====	2859
!	2860
! VARIABLE ASSOCIATED WITH SYSTEM FOR WHICH FAILURE INFORMATION	2861
! PROPAGATION DATA IS CURRENTLY BEING ENTERED	2862
!	2863
=====	2864
!	2865
DECLARE TSYSTEM PIC X(4).	2866
!	2867
!	2868
=====	2869
!	2870
! VARIABLES ASSOCIATED WITH FMCODE	2871
!	2872
=====	2873
!	2874
DECLARE TFMCODE PIC X(20).	2875
DECLARE TSSYS PIC X(4).	2876
DECLARE TSSYSMOD PIC X(8).	2877
!	2878
!	2879
=====	2880
!	2881
! VARIABLES ASSOCIATED WITH CODE_NUMBER	2882
!	2883
=====	2884
!	2885
DECLARE TCODENO PIC X(21).	2886
DECLARE TTCODENO PIC X(21).	2887
DECLARE TSYSA PIC X(4).	2888
DECLARE TSYSMODA PIC X(8).	2889
DECLARE TTSYSMODA PIC X(8).	2890
DECLARE TCONNECTU PIC X(5).	2891
DECLARE TSYSB PIC X(4).	2892
DECLARE TSYSMODB PIC X(8).	2893
DECLARE TTSYSMODB PIC X(8).	2894
!	2895
!	2896
=====	2897
!	2898
! VARIABLES ASSOCIATED WITH OTHER INPUT FIELDS FOR	2899
! DOMAIN PROPAGATIONS:	2900
! 1. SIGNAL TYPE	2901
! 2. DIMENSIONS	2902
! 3. SIGNAL QUALITY	2903
! 4. MAX_FREQ_OR_TIME	2904
! 5. MIN_FREQ_OR_TIME	2905
! 6. PARAMETER	2906

Datatrieve Procedure FIP_STORE (cont.)

!	7. SYMPTOM_DURATION	2907
!	8. PERIOD_OF_ONSET	2908
!	9. INDICATES_FAILURE	2909
!	10. COMMENT1	2910
!	11. COMMENT2	2911
!	12. COMMENT3	2912
!		2913
!	=====	2914
!		2915
DECLARE	TSIG PIC X(20).	2916
DECLARE	TDIM PIC 9(1).	2917
DECLARE	TSIGQUAL PIC 9(1).	2918
DECLARE	TMAXFT PIC S9(2).	2919
DECLARE	TMINFT PIC S9(2).	2920
DECLARE	TPAR PIC X(20).	2921
DECLARE	TSYMDUR PIC S9(2).	2922
DECLARE	TPDONSET PIC S9(2).	2923
DECLARE	TINDFAIL PIC X(1).	2924
DECLARE	TCOMMENT1 PIC X(80).	2925
DECLARE	TCOMMENT2 PIC X(80).	2926
DECLARE	TCOMMENT3 PIC X(80).	2927
DECLARE	TCOMMENT PIC X(80).	2928
!		2929
!		2930
!	=====	2931
!	VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR COMMENTS	2932
!		2933
!	=====	2934
!		2935
DECLARE	TEMP PIC X(249).	2936
!		2937
!		2938
!		2939
!	=====	2940
!	VARIABLES USED AS FLAGS OR CONDITION INDICATORS	2941
!		2942
!	=====	2943
!		2944
!		2945
DECLARE	TCONTINUE1 PIC X(1).	2946
DECLARE	TCONTINUE2 PIC X(1).	2947
DECLARE	TCONTINUE3 PIC X(1).	2948
DECLARE	TFAILSAME PIC X(1).	2949
DECLARE	IMSG PIC 9(2).	2950
DECLARE	FLG1 PIC X(1).	2951
DECLARE	FLG2 PIC X(1).	2952
!		2953
!		2954
!	=====	2955
!	VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM	2956
!		2957

Datatrieve Procedure FIP_STORE (cont.)

```

!
!=====
!
!
DECLARE TMSG      PIC X(80).
DECLARE TMSG1     PIC X(80).
DECLARE TMSG2     PIC X(80).
DECLARE TMSG3     PIC X(80).
DECLARE TMSG4     PIC X(80).
DECLARE TMSG5     PIC X(80).
DECLARE TMSG6     PIC X(80).
DECLARE TMSG7     PIC X(80).
DECLARE TMSG8     PIC X(80).
DECLARE TMSG9     PIC X(80).
DECLARE TMSG10    PIC X(80).
DECLARE TMSG11    PIC X(80).
DECLARE TMSG12    PIC X(80).
DECLARE TMSG13    PIC X(80).
DECLARE TMSG14    PIC X(80).
!
!=====
!
! VARIABLE USED AS COUNTERS
!=====
!
DECLARE ICNT      PIC 9(4).
DECLARE JCNT      PIC 9(4).
DECLARE I         PIC 9(4).
DECLARE J         PIC 9(4).
!
!=====
!
! VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE
!=====
!
DECLARE CAL      USAGE DATE
                EDIT_STRING X(23).
!
!=====
!
! READY THE DOMAINS CONNECTIONS, MODULES, SYSTEMS AND PROPAGATIONS_FORM
!=====
!
SET ABORT
READY CONNECTIONS      SHARED READ
READY FAILUREMODES     SHARED READ

```

Datatrieve Procedure FIP_STORE (cont.)

```

READY SYSTEMS          SHARED WRITE          3009
READY PROPAGATIONS_FORM SHARED READ          3010
!                                                    3011
!                                                    3012
!=====                                                    3013
!                                                    3014
! INITIALIZE THE MESSAGE VARIABLES, INITIALIZE THE COUNTER (ICNT) USED  3015
! FOR NUMBERING THE LOG FILE RECORDS AND INITIALIZE THE COUNTER (JCNT)  3016
! USED TO CONTROL THE NUMBER OF RECORDS PRINTED ON EACH PAGE OF THE LOG  3017
! FILE                                                    3018
!=====                                                    3019
!                                                    3020
!                                                    3021
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"      3022
TMSG2 = "CODE NUMBER IS NOT VALID -- SYSTEM A OR SYSTEM B MUST BE"    3023
TMSG3 = "CODE NUMBER IS NOT VALID -- NOT IN DOMAIN CONNECTIONS"        3024
TMSG4 = "SIGNAL QUALITY IS NOT VALID -- MUST BE 0, 1, 2, 3, 4 OR 5"    3025
TMSG5 = "FMCODE IS NOT VALID -- SOURCE SYSTEM MUST BE"                 3026
TMSG6 = "FMCODE IS NOT VALID -- NOT IN DOMAIN FAILUREMODES"            3027
TMSG7 = "SIGNAL TYPE IS NOT VALID -- NOT IN TABLE SIGNAL_TABLE"       3028
TMSG8 = "DIMENSIONS IS NOT VALID -- MUST BE 1, 2 OR 3"                 3029
TMSG9 = "PARAMETER IS NOT VALID -- NOT IN TABLE PARAMETER_TABLE"      3030
TMSG10 = "INDICATES FAILURE IS NOT VALID -- MUST BE T OR F"            3031
TMSG11 = "PROPAGATION RECORD IS NOT VALID FOR THIS FMCODE, " |        3032
        "SIGNAL TYPE AND PARAMETER"                                    3033
TMSG12 = "PROPAGATION RECORD IS NOT VALID -- ALREADY EXISTS IN " |     3034
        "DOMAIN PROPAGATIONS "                                         3035
TMSG13 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"                  3036
TMSG14 = "MORE DATA THIS FAILURE MODE/SIGNAL IS NOT VALID -- MUST BE " | 3037
        "Y OR N"                                                        3038
ICNT = 0                                                            3039
JCNT = 0                                                            3040
!                                                    3041
!                                                    3042
!=====                                                    3043
!                                                    3044
! THIS SECTION CALLS PROCEDURE CREATE_PROPAGATIONS_FIP_1 TO ENSURE      3045
! THAT A PROPAGATIONS DOMAIN AND DATA FILE EXIST FOR EACH SYSTEM IN    3046
! DOMAIN SYSTEMS                                                         3047
!=====                                                    3048
!                                                    3049
!=====                                                    3050
:CREATE_PROPAGATIONS_FIP_1                                           3051
FN$DELETE_LOG("PROC")                                              3052
!                                                    3053
!                                                    3054
!=====                                                    3055
!                                                    3056
! THIS SECTION CALLS PROCEDURE FIP_STORE_1 TO REQUEST A SYSTEM, READY   3057
! THE CORRESPONDING DOMAIN AND CALL PROCEDURE FIP_STORE_2 TO REQUEST     3058
! ENTRY OF FAILURE INFORMATION PROPAGATIONS FOR THE DESIGNATED SYSTEM   3059

```

Datatrieve Procedure FIP_STORE (cont.)

!	3060
!-----	3061
!	3062
:FIP_STORE_1	3063
FN\$DELETE_LOG("PROPAGATIONS")	3064
FN\$DELETE_LOG("PROPAGATIONS_FILE")	3065
FN\$DELETE_LOG("PROC")	3066
END-PROCEDURE	3067

Datatrieve Procedure FIP STORE 1

```

DEFINE PROCEDURE FIP_STORE_1
!
!
!=====
!
! LOOP TO REQUEST SYSTEM FOR WHICH FAILURE INFORMATION PROPAGATIONS
! ARE TO BE ENTERED AND VERIFY THAT THIS SYSTEM EXISTS IN DOMAIN SYSTEMS
!=====
!
FLG1      = "N"
TCONTINUE1 = "Y"
WHILE FLG1 = "N" AND
      TCONTINUE1 = "Y"
BEGIN
  PRINT NEW_PAGE
  :CLRSCRN
  PRINT SKIP 3,
    "===== STORE FAILURE INFORMATION PROPAGATIONS =====",
    SKIP 1
  TSYSTEM = FN$UPCASE(*."SYSTEM")
  FOR SYSTEMS WITH SYSTEM = TSYSTEM
    FLG1 = "Y"
  IF FLG1 = "N" THEN
    BEGIN
      PRINT SKIP 2, "System", SPACE 1, TSYSTEM (-) USING X(4),
        SPACE 1, "is not defined in domain SYSTEMS.", SKIP 1
      TCONTINUE1 = "X"
      WHILE TCONTINUE1 NE "Y" AND
        TCONTINUE1 NE "N"
        BEGIN
          PRINT SKIP 1, "Do you wish to continue?", SKIP 1
          TCONTINUE1 = FN$UPCASE(*."Y or N")
          PRINT " "
        END
      IF TCONTINUE1 = "N" THEN
        BEGIN
          CAL = "NOW"
          PRINT NEW_PAGE
          :CLRSCRN
          PRINT SKIP 2,
            " STORE PROPAGATIONS", SKIP 2,
            "===== ", SKIP 2,
            "===== END: ", SPACE 0,
            CAL (-) USING X(17),
            SPACE 0, " =====", SKIP 2,
            "===== ", SKIP 3
          ABORT
        END
      END
    END
  END
END

```


Datatrieve Procedure FIP_STORE_1 (cont.)

!	3119
!	3120
!=====	3121
!	3122
! DATATRIEVE LOGICALS ARE CREATED TO REPRESENT THE NAMES OF THE	3123
! PROPAGATIONS DOMAIN AND DATA FILE ASSOCIATED WITH THE CURRENT	3124
! SYSTEM. THE DOMAIN IS NAMED "PROPAGATIONS_X999" WHERE "X999"	3125
! IS THE CURRENT SYSTEM. THE DATA FILE IS NAMED	3126
! "DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_X999.DAT"	3127
!=====	3128
!	3129
!	3130
IF FN\$TRANS LOG("PROPAGATIONS") NE "PROPAGATIONS" THEN	3131
FN\$DELETE LOG("PROPAGATIONS")	3132
FN\$CREATE LOG("PROPAGATIONS", "PROPAGATIONS " TSYSTEM)	3133
IF FN\$TRANS LOG("PROPAGATIONS_FILE") NE "PROPAGATIONS_FILE" THEN	3134
FN\$DELETE LOG("PROPAGATIONS_FILE")	3135
FN\$CREATE LOG("PROPAGATIONS_FILE",	3136
"DEV\$206:[BCDSSME2.DATA]PROPAGATIONS_" TSYSTEM ".DAT")	3137
!	3138
!	3139
!=====	3140
!	3141
! READY THE PROPAGATIONS DOMAIN FOR THE CURRENT SYSTEM	3142
!	3143
!=====	3144
!	3145
READY PROPAGATIONS SHARED WRITE	3146
!	3147
!	3148
!=====	3149
!	3150
! PROCEDURE FIP_STORE 2 IS CALLED TO REQUEST INPUT OF FAILURE	3151
! INFORMATION PROPAGATIONS FOR THE CURRENT SYSTEM	3152
!	3153
!=====	3154
!	3155
:FIP_STORE 2	3156
END-PROCEDURE	3157

Datatrieve Procedure FIP_STORE_2

```

DEFINE PROCEDURE FIP_STORE_2                                     3158
!                                                                3159
!                                                                3160
!=====                                                       3161
!                                                                3162
! PRIMARY LOOP TO STORE PROPAGATIONS                             3163
!                                                                3164
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:                3165
! 1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR  3166
!    CORRECTION OF INVALID INFORMATION                           3167
! 2. IF TCONTINUE2 NE "A", SECTION TO REARRANGE THE VARIABLES  3168
!    ASSOCIATED WITH COMMENTS                                   3169
! 3. IF TCONTINUE2 NE "A", SECTION TO STORE RECORD IN DOMAIN    3170
!    PROPAGATIONS X999 AFTER VALIDATION TESTS HAVE BEEN PASSED 3171
!    ("X999" REPRESENTS THE CURRENT SYSTEM BEING PROCESSED)    3172
! 4. IF TCONTINUE2 NE "A", SECTION TO PRINT DATA STORED IN DOMAIN 3173
!    PROPAGATIONS X999 FOR INCLUSION IN THE SESSION LOG FILE    3174
! 5. IF TCONTINUE2 = "A", SECTION TO PRINT MESSAGE THAT DATA   3175
!    CURRENTLY ON FORM HAS NOT BEEN STORED                       3176
! 6. IF TCONTINUE2 = "A", SECTION TO REQUEST RESPONSE TO CONTINUE 3177
!    ENTERING DATA FOR CURRENT SYSTEM                           3178
! 7. SECTION TO REQUEST RESPONSE TO CONTINUE PROCEDURE OR EXIT TO 3179
!    MENU                                                         3180
!=====                                                       3181
!                                                                3182
! TCONTINUE2 = "Y"                                              3183
! TFAILSAME = "N"                                              3184
! WHILE TCONTINUE2 = "Y"                                       3185
! BEGIN                                                         3186
!                                                                3187
!                                                                3188
!=====                                                       3189
!                                                                3190
! LOOP TO DISPLAY BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON 3191
! THE FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF    3192
! INVALID DATA                                                  3193
!=====                                                       3194
!                                                                3195
!=====                                                       3196
!                                                                3197
! FLG1 = "N"                                                    3198
! IMSG = 1                                                       3199
! WHILE FLG1 = "N"                                              3200
! BEGIN                                                         3201
!     IF IMSG = 1 THEN TMSG = TMSG1                             3202
!     IF IMSG = 2 THEN TMSG = TMSG2 || " " | TSYSTEM           3203
!     IF IMSG = 3 THEN TMSG = TMSG3                             3204
!     IF IMSG = 4 THEN TMSG = TMSG4                             3205
!     IF IMSG = 5 THEN TMSG = TMSG5 || " " | TSYSTEM           3206
!     IF IMSG = 6 THEN TMSG = TMSG6                             3207
!     IF IMSG = 7 THEN TMSG = TMSG7                             3208

```

Datatrieve Procedure FIP_STORE_2 (cont.)

```

IF IMMSG = 8 THEN TMSG = TMSG8                                3209
IF IMMSG = 9 THEN TMSG = TMSG9                                3210
IF IMMSG = 10 THEN TMSG = TMSG10                              3211
IF IMMSG = 11 THEN TMSG = TMSG11                              3212
IF IMMSG = 12 THEN TMSG = TMSG12 || TSYSTEM                   3213
IF IMMSG = 13 THEN TMSG = TMSG13                              3214
IF IMMSG = 14 THEN TMSG = TMSG14                              3215
                                                                3216
                                                                3217
=====                                                       3218
THIS SECTION DISPLAYS THE STORE FAILURE INFORMATION           3219
PROPAGATIONS FORM AND RETRIEVES THE DATA ENTERED ON THE     3220
FORM                                                           3221
=====                                                       3222
                                                                3223
                                                                3224
                                                                3225
FOR FIRST 1 PROPAGATIONS_FORM                                3226
  BEGIN                                                       3227
    DISPLAY FORM PROPAGATIONS STO FORM IN                     3228
      DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING             3229
      BEGIN                                                    3230
        IF IMMSG NE 1 THEN                                     3231
          BEGIN                                                 3232
            PUT_FORM CODE_NUMBER = TCODENO                     3233
            PUT_FORM QUALITY = TSIGQUAL                         3234
            PUT_FORM FAILSAME = TFAILSAME                       3235
          END                                                    3236
          IF IMMSG = 1 AND                                       3237
            TFAILSAME = "N" THEN                                3238
            BEGIN                                                3239
              PUT_FORM FMCODE = "_____----0000"             3240
            END                                                    3241
          IF IMMSG NE 1 OR                                       3242
            (IMMSG = 1 AND                                       3243
              TFAILSAME = "Y") THEN                              3244
            BEGIN                                                3245
              PUT_FORM FMCODE = TFMCODE                         3246
              PUT_FORM SIGNAL_TYPE = TSIG                       3247
              PUT_FORM DIMENSIONS = TDIM                        3248
              PUT_FORM MAX_FREQ_TIME = TMAXFT                   3249
              PUT_FORM MIN_FREQ_TIME = TMINFT                   3250
              PUT_FORM PARAM = TPAR                             3251
              PUT_FORM DURATION = TSYMDUR                       3252
              PUT_FORM ONSET = TPDONSET                         3253
              PUT_FORM FAILURE = TINDFAIL                       3254
              PUT_FORM COMMENT_1 = TCOMMENT1                    3255
              PUT_FORM COMMENT_2 = TCOMMENT2                    3256
              PUT_FORM COMMENT_3 = TCOMMENT3                    3257
            END                                                    3258
          IF IMMSG = 1 THEN                                      3259

```

Datatrieve Procedure FIP_STORE_2 (cont.)

```

      BEGIN                                     3260
        PUT_FORM FAILSAME = "Y"               3261
      END                                     3262
      PUT_FORM CONTINUE = TCONTINUE2          3263
      PUT_FORM MESSAGE = TMSG                 3264
    END RETRIEVE USING                        3265
    BEGIN                                     3266
      TCODENO = GET_FORM CODE_NUMBER          3267
      TSYSMODA = FN$STR_EXTRACT(TCODENO, 1,8)  3268
      TCONNECTU = FN$STR_EXTRACT(TCODENO, 9,5) 3269
      TSYSMODB = FN$STR_EXTRACT(TCODENO,14,8)  3270
      IF TSYSMODA > TSYSMODB THEN              3271
        BEGIN                                 3272
          TCODENO = TSYSMODB || TCONNECTU || TSYSMODA 3273
        END                                     3274
      TFMCODE = GET_FORM FMCODE               3275
      TSIG = GET_FORM SIGNAL_TYPE              3276
      TDIM = GET_FORM DIMENSIONS               3277
      TSIGQUAL = GET_FORM QUALITY              3278
      TMAXFT = GET_FORM MAX_FREQ_TIME          3279
      TMINFT = GET_FORM MIN_FREQ_TIME          3280
      IF TMAXFT < TMINFT THEN                  3281
        BEGIN                                 3282
          TMAXFT = GET_FORM MIN_FREQ_TIME        3283
          TMINFT = GET_FORM MAX_FREQ_TIME        3284
        END                                     3285
      TPAR = GET_FORM PARAM                    3286
      TSYMDUR = GET_FORM DURATION              3287
      TPDONSET = GET_FORM ONSET                3288
      TINDFAIL = GET_FORM FAILURE              3289
      TCOMMENT1 = GET_FORM COMMENT_1           3290
      TCOMMENT2 = GET_FORM COMMENT_2           3291
      TCOMMENT3 = GET_FORM COMMENT_3           3292
      TCONTINUE2 = GET_FORM CONTINUE           3293
      TFAILSAME = GET_FORM FAILSAME            3294
    END                                     3295
  END                                     3296
!                                     3297
!                                     3298
!                                     3299
!                                     3300
!                                     3301
!                                     3302
!                                     3303
!                                     3304
!                                     3305
!                                     3306
!                                     3307
!                                     3308
!                                     3309
!                                     3310

```

IF TCONTINUE2 IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH WERE INADVERTENTLY ENTERED IN TSIG, TPAR, TCOMMENT1, TCOMMENT2 AND TCOMMENT3 ARE REMOVED

```

    IF TCONTINUE2 NE "A" THEN
      BEGIN
        I = 1
        WHILE I LE 5

```

Datatrieve Procedure FIP_STORE_2 (cont.)

```

BEGIN                                                    3311
  IF I = 1 THEN TEMP = TSIG                             3312
  IF I = 2 THEN TEMP = TPAR                             3313
  IF I = 3 THEN TEMP = TCOMMENT1                       3314
  IF I = 4 THEN TEMP = TCOMMENT2                       3315
  IF I = 5 THEN TEMP = TCOMMENT3                       3316
  IF TEMP NE " " THEN                                  3317
    BEGIN                                              3318
      J = 1                                           3319
      WHILE FN$STR_EXTRACT(TEMP, J, 1) = " "         3320
        BEGIN                                         3321
          J = J + 1                                   3322
        END                                           3323
      TEMP = FN$STR_EXTRACT(TEMP, J, 249 - J + 1)     3324
    END                                              3325
    IF I = 1 THEN TSIG      = TEMP                   3326
    IF I = 2 THEN TPAR      = TEMP                   3327
    IF I = 3 THEN TCOMMENT1 = TEMP                   3328
    IF I = 4 THEN TCOMMENT2 = TEMP                   3329
    IF I = 5 THEN TCOMMENT3 = TEMP                   3330
    I = I + 1                                         3331
  END                                                3332
END                                                  3333
!                                                    3334
!                                                    3335
===== 3336
! IF TCONTINUE2 IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO 3337
! VERIFY THE FOLLOWING: 3338
! 1. TSYSA OR TSYSB IS EQUAL TO TSYSTEM 3339
! 2. TCODENO IS IN DOMAIN CONNECTIONS 3340
! 3. TSIGQUAL IS 0, 1, 2, 3, 4 OR 5 3341
! 4. TSSYS IS EQUAL TO TSYSTEM 3342
! 5. TFMCODE IS IN DOMAIN FAILUREMODES 3343
! 6. TSIG IS IN TABLE SIGNAL_TABLE 3344
! 7. TDIM IS 1, 2 OR 3 3345
! 8. TPAR IS IN TABLE PARAMETER_TABLE 3346
! 9. TINDFAIL IS "T" OR "F" 3347
! 10. TFMCODE, TSIG AND TPAR HAVE A VALID PROPAGATION PATH 3348
! TO CONNECTION = TCODENO 3349
! 11. TCODENO, TFMCODE, TSIG AND TPAR DO NO ALREADY EXIST IN 3350
! DOMAIN PROPAGATIONS_X999 WHERE "X999" REPRESENTS THE 3351
! CURRENT SYSTEM 3352
! 12. TCONTINUE2 IS "Y", "N" OR "A" 3353
! 13. TFAILSAME IS "Y" OR "N" 3354
===== 3355
!                                                    3356
!                                                    3357
!                                                    3358
FLG1 = "Y" 3359
IF TCONTINUE2 NE "A" THEN 3360
  BEGIN 3361

```

Datatrieve Procedure FIP_STORE_2 (cont.)

TSYSA = FN\$STR_EXTRACT(TCODENO, 1, 4)	3362
TSYSB = FN\$STR_EXTRACT(TCODENO, 14, 4)	3363
IF TSYSA NE TSYSYSTEM AND	3364
TSYSB NE TSYSYSTEM THEN	3365
BEGIN	3366
FLG1 = "N"	3367
IMSG = 2	3368
END	3369
IF FLG1 = "Y" THEN	3370
BEGIN	3371
FLG2 = "N"	3372
FOR CONNECTIONS WITH CODE_NUMBER = TCODENO	3373
BEGIN	3374
FLG2 = "Y"	3375
TTSYSMODA = SYSTEM_MODULE_A	3376
TTSYSMODB = SYSTEM_MODULE_B	3377
END	3378
IF FLG2 = "N" THEN	3379
BEGIN	3380
FLG1 = "N"	3381
IMSG = 3	3382
END	3383
END	3384
IF FLG1 = "Y" THEN	3385
BEGIN	3386
IF TSIGQUAL NE 0 AND	3387
TSIGQUAL NE 1 AND	3388
TSIGQUAL NE 2 AND	3389
TSIGQUAL NE 3 AND	3390
TSIGQUAL NE 4 AND	3391
TSIGQUAL NE 5 THEN	3392
BEGIN	3393
FLG1 = "N"	3394
IMSG = 4	3395
END	3396
END	3397
IF FLG1 = "Y" THEN	3398
BEGIN	3399
TSSYS = FN\$STR_EXTRACT(TFMCODE, 1, 4)	3400
IF TSSYS NE TSYSYSTEM THEN	3401
BEGIN	3402
FLG1 = "N"	3403
IMSG = 5	3404
END	3405
END	3406
IF FLG1 = "Y" THEN	3407
BEGIN	3408
FLG2 = "N"	3409
FOR FAILUREMODES WITH FMCODE = TFMCODE	3410
BEGIN	3411
FLG2 = "Y"	3412

Datatrieve Procedure FIP_STORE_2 (cont.)

TSSYSMOD = SOURCE_SYSTEM_MODULE	3413
END	3414
IF FLG2 = "N" THEN	3415
BEGIN	3416
FLG1 = "N"	3417
IMSG = 6	3418
END	3419
END	3420
IF FLG1 = "Y" THEN	3421
BEGIN	3422
IF TSIG NOT IN SIGNAL_TABLE THEN	3423
BEGIN	3424
FLG1 = "N"	3425
IMSG = 7	3426
END	3427
END	3428
IF FLG1 = "Y" THEN	3429
BEGIN	3430
IF TDIM NE 1 AND	3431
TDIM NE 2 AND	3432
TDIM NE 3 THEN	3433
BEGIN	3434
FLG1 = "N"	3435
IMSG = 8	3436
END	3437
END	3438
IF FLG1 = "Y" THEN	3439
BEGIN	3440
IF TPAR NOT IN PARAMETER_TABLE THEN	3441
BEGIN	3442
FLG1 = "N"	3443
IMSG = 9	3444
END	3445
END	3446
IF FLG1 = "Y" THEN	3447
BEGIN	3448
IF TINDFAIL NE "T" AND	3449
TINDFAIL NE "F" THEN	3450
BEGIN	3451
FLG1 = "N"	3452
IMSG = 10	3453
END	3454
END	3455
IF FLG1 = "Y" THEN	3456
BEGIN	3457
FLG2 = "N"	3458
IF TSSYSMOD = TSSYSMODA OR	3459
TSSYSMOD = TSSYSMODB THEN	3460
BEGIN	3461
FLG2 = "Y"	3462
END	3463

Datatrieve Procedure FIP_STORE_2 (cont.)


```

3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565

```

IF TFAILSAME NE "Y" AND	3515
TFailsame NE "N" THEN	3516
BEGIN	3517
FLG1 = "N"	3518
IMSG = 14	3519
END	3520
END	3521
END	3522
END	3523
=====	
IF TCONTINUE2 IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE	3527
VARIABLES TCOMMENT1, TCOMMENT2 AND TCOMMENT3 SO THAT THE FIRST	3528
VALUE WHICH IS NOT BLANK IS TCOMMENT1 AND THE SECOND VALUE WHICH	3529
IS NOT BLANK IS TCOMMENT2	3530
=====	
IF TCONTINUE2 NE "A" THEN	3535
BEGIN	3536
TEMP = TCOMMENT1 "\$#" TCOMMENT2 "\$#" TCOMMENT3 "\$#"	3537
I = 1	3538
WHILE I LE 3	3539
BEGIN	3540
J = 1	3541
WHILE J = 1	3542
BEGIN	3543
J = FN\$STR_LOC(TEMP, "\$#")	3544
IF J = 0 THEN TCOMMENT = " "	3545
IF J = 1 THEN TEMP = FN\$STR_EXTRACT(TEMP, 4, 246)	3546
IF J GT 1 THEN	3547
BEGIN	3548
TCOMMENT = FN\$STR_EXTRACT(TEMP, 1, J - 1)	3549
TEMP = FN\$STR_EXTRACT(TEMP, J, 249 - J + 1)	3550
END	3551
END	3552
IF I = 1 THEN TCOMMENT1 = TCOMMENT	3553
IF I = 2 THEN TCOMMENT2 = TCOMMENT	3554
IF I = 3 THEN TCOMMENT3 = TCOMMENT	3555
I = I + 1	3556
END	3557
END	3558
END	3559
=====	
IF TCONTINUE2 IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN	3562
DOMAIN PROPAGATIONS_X999 WHERE "X999" REPRESENTS THE CURRENT SYSTEM	3563

Datatrieve Procedure FIP_STORE_2 (cont.)

```

===== 3566
! 3567
! IF TCONTINUE2 NE "A" THEN 3568
! BEGIN 3569
!     CAL = "NOW" 3570
!     STORE PROPAGATIONS USING 3571
!     BEGIN 3572
!         DATE_CREATED = CAL 3573
!         FMCODE = TFMCODE 3574
!         CODE_NUMBER = TCODENO 3575
!         SIGNAL_TYPE = TSIG 3576
!         SIGNAL_UNITS = TSIG VIA SIGNAL_TABLE 3577
!         DIMENSIONS = TDIM 3578
!         SIGNAL_QUALITY = TSIGQUAL 3579
!         MAX_FREQ_OR_TIME = TMAXFT 3580
!         MIN_FREQ_OR_TIME = TMINFT 3581
!         FT_UNITS = TSIG VIA FREQ_TIME_UNITS_TABLE 3582
!         PARAMETER = TPAR 3583
!         PARAMETER_UNITS = TPAR VIA PARAMETER_TABLE 3584
!         SYMPTOM_DURATION = TSYMDUR 3585
!         PERIOD_OF_ONSET = TPDONSET 3586
!         INDICATES_FAILURE = TINDFAIL 3587
!         COMMENT1 = TCOMMENT1 3588
!         COMMENT2 = TCOMMENT2 3589
!         COMMENT3 = TCOMMENT3 3590
!     END 3591
! END 3592
! 3593
! 3594
! 3595
! ===== 3596
! IF TCONTINUE2 IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN 3597
! PROPAGATIONS X999 IS PRINTED. THE OUTPUT OF THE PRINT STATEMENTS 3598
! WILL BE INCLUDED IN THE SESSION LOG FILE WHICH IS OPENED BY THE 3599
! CALLING COMMAND PROCEDURE. 3600
! ===== 3601
! 3602
! 3603
! IF TCONTINUE2 NE "A" THEN 3604
! BEGIN 3605
!     ICNT = ICNT + 1 3606
!     JCNT = JCNT + 1 3607
!     IF JCNT = 1 THEN PRINT NEW PAGE 3608
!     FOR PROPAGATIONS WITH FMCODE = TFMCODE AND 3609
!                             CODE_NUMBER = TCODENO AND 3610
!                             SIGNAL_TYPE = TSIG AND 3611
!                             PARAMETER = TPAR 3612
!     PRINT SKIP 2, 3613
!     COL 1, "RECORD NO.", SPACE 1, 3614
!     ICNT (-) USING ZZ9, SKIP 1, 3615
!     COL 1, "=====", SPACE 0, 3616

```

Datatrieve Procedure FIP_STORE_2 (cont.)

```

"=====", SPACE 0, 3617
"=====", SPACE 0, 3618
"=====", SKIP 2, 3619
COL 3, "DATE CREATED      :", SPACE 1, 3620
CAL (-) USING X(23), SKIP 1, 3621
COL 3, "FMCODE            :", SPACE 1, 3622
TFMCODE (-) USING X(20), SKIP 1, 3623
COL 3, "CODE NUMBER       :", SPACE 1, 3624
TCODENO (-) USING X(21), SKIP 1, 3625
COL 3, "SIGNAL TYPE        :", SPACE 1, 3626
TSIG (-) USING X(20), SKIP 1, 3627
COL 3, "SIGNAL UNITS       :", SPACE 1, 3628
(TSIG VIA SIGNAL_TABLE) (-) USING X(25), SKIP 1, 3629
COL 3, "DIMENSIONS        :", SPACE 1, 3630
TDIM (-) USING 9(1), SKIP 1, 3631
COL 3, "SIGNAL QUALITY     :", SPACE 1, 3632
TSIGQUAL (-) USING 9(1), SKIP 1, 3633
COL 3, "MAX FREQ OR TIME   :", SPACE 1, 3634
TMAXFT (-) USING -Z9, SKIP 1, 3635
COL 3, "MIN FREQ OR TIME   :", SPACE 1, 3636
TMINFT (-) USING -Z9, SKIP 1, 3637
COL 3, "FT UNITS           :", SPACE 1, 3638
(TSIG VIA FREQ_TIME_UNITS_TABLE) (-) USING X(25), 3639
SKIP 1, 3640
COL 3, "PARAMETER          :", SPACE 1, 3641
TPAR (-) USING X(20), SKIP 1, 3642
COL 3, "PARAMETER UNITS    :", SPACE 1, 3643
(TPAR VIA PARAMETER_TABLE) (-) USING X(25), SKIP 1, 3644
COL 3, "SYMPTOM DURATION   :", SPACE 1, 3645
TSYMDUR (-) USING -Z9, SKIP 1, 3646
COL 3, "PERIOD OF ONSET    :", SPACE 1, 3647
TPDONSET (-) USING -Z9, SKIP 1, 3648
COL 3, "INDICATES FAILURE  :", SPACE 1, 3649
TINDFAIL (-) USING X(1), SKIP 1, 3650
COL 3, "COMMENT1           :", SPACE 1, 3651
TCOMMENT1 (-) USING T(50), SKIP 1, 3652
COL 3, "COMMENT2           :", SPACE 1, 3653
TCOMMENT2 (-) USING T(50), SKIP 1, 3654
COL 3, "COMMENT3           :", SPACE 1, 3655
TCOMMENT3 (-) USING T(50), SKIP 2, 3656
COL 1, "=====", SPACE 0, 3657
"=====", SPACE 0, 3658
"=====", SPACE 0, 3659
"=====" 3660
IF JCNT = 2 THEN JCNT = 0 3661
:BELL 3662
END 3663
! 3664
! 3665
! ===== 3666
! | 3667

```

Datatrieve Procedure FIP_STORE_2 (cont.)

```

!      IF TCONTINUE2 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE
!      THAT THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED.
!      THIS MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE.
!      =====
!
!      IF TCONTINUE2 = "A" THEN
!      BEGIN
!      JCNT = JCNT + 1
!      IF JCNT = 1 THEN PRINT NEW_PAGE
!      PRINT SKIP 3,
!      COL 1, "=====", SPACE 0,
!      "=====", SPACE 0,
!      "=====", SPACE 0,
!      "=====", SKIP 7,
!      COL 9, "*****", SKIP 2,
!      COL 9, "***** RECORD NOT STORED *****", SKIP 2,
!      COL 9, "*****", SKIP 7,
!      COL 1, "=====", SPACE 0,
!      "=====", SPACE 0,
!      "=====", SPACE 0,
!      "====="
!      IF JCNT = 2 THEN JCNT = 0
!      END
!
!      =====
!
!      IF TCONTINUE2 EQUALS "A", A RESPONSE IS REQUESTED TO CONTINUE
!      ENTERING FAILURE INFORMATION PROPAGATIONS FOR SYSTEM = TSYSTEM
!      =====
!
!      IF TCONTINUE2 = "A" THEN
!      BEGIN
!      PRINT NEW_PAGE
!      :CLRSCRN
!      TCONTINUE2 = "X"
!      WHILE TCONTINUE2 NE "Y" AND
!      TCONTINUE2 NE "N"
!      BEGIN
!      PRINT SKIP 3,
!      "Do you wish to continue entering", SKIP 1,
!      "FAILURE INFORMATION PROPAGATIONS", SKIP 1,
!      "for system", SPACE 1, TSYSTEM (-) USING X(4),
!      SPACE 0, "?", SKIP 1
!      TCONTINUE2 = FNSUPCASE(*."Y or N")
!      PRINT " "
!      END
!      TFAILSAME = "N"
!      END

```

Datatrieve Procedure FIP_STORE_2 (cont.)

END	3719
!	3720
!	3721
=====	3722
!	3723
! SECTION TO REQUEST RESPONSE TO CONTINUE ENTERING FAILURE INFORMATION	3724
! PROPAGATIONS FOR A NEW SYSTEM OR TO EXIT PROCEDURE. IF TCONTINUE3 IS	3725
! EQUAL TO "Y", PROCEDURE FIP_STORE_1 IS CALLED TO REQUEST A NEW SYSTEM.	3726
! IF TCONTINUE3 IS EQUAL TO "N", PROCEDURE DTR_NULL IS CALLED TO	3727
! TERMINATE STORING FAILURE INFORMATION PROPAGATIONS	3728
!	3729
=====	3730
!	3731
PRINT NEW_PAGE	3732
:CLRSCRN	3733
TCONTINUE3 = "X"	3734
WHILE TCONTINUE3 NE "Y" AND	3735
TCONTINUE3 NE "N"	3736
BEGIN	3737
PRINT SKIP 3,	3738
"Do you wish to continue entering", SKIP 1,	3739
"FAILURE INFORMATION PROPAGATIONS", SKIP 1,	3740
"for another system?", SKIP 1	3741
TCONTINUE3 = FN\$UPCASE(*."Y or N")	3742
PRINT " "	3743
END	3744
IF FN\$TRANS_LOG("PROC") NE "PROC" THEN	3745
FN\$DELETE_LOG("PROC")	3746
CHOICE	3747
TCONTINUE3 = "Y" THEN FN\$CREATE_LOG("PROC", "FIP_STORE_1")	3748
TCONTINUE3 = "N" THEN FN\$CREATE_LOG("PROC", "DTR_NULL")	3749
END CHOICE	3750
:PROC	3751
END-PROCEDURE	3752

Datatrieve Procedure FM MODIFY

Datatrieve Procedure FM_MODIFY (cont.)

! VARIABLES ASSOCIATED WITH EFFECTS	3804
! =====	3805
! =====	3806
! =====	3807
DECLARE TEFFECT1A PIC X(80).	3808
DECLARE TEFFECT1B PIC X(80).	3809
DECLARE TEFFECT2A PIC X(80).	3810
DECLARE TEFFECT2B PIC X(80).	3811
DECLARE TEFFECT3A PIC X(80).	3812
DECLARE TEFFECT3B PIC X(80).	3813
DECLARE TEFFECT4A PIC X(80).	3814
DECLARE TEFFECT4B PIC X(80).	3815
DECLARE TEFFECT5A PIC X(80).	3816
DECLARE TEFFECT5B PIC X(80).	3817
DECLARE TEFFECT6A PIC X(80).	3818
DECLARE TEFFECT6B PIC X(80).	3819
DECLARE TEFFECT1 PIC X(161).	3820
DECLARE TEFFECT2 PIC X(161).	3821
DECLARE TEFFECT3 PIC X(161).	3822
DECLARE TEFFECT4 PIC X(161).	3823
DECLARE TEFFECT5 PIC X(161).	3824
DECLARE TEFFECT6 PIC X(161).	3825
DECLARE TTEFFECT1A PIC X(80).	3826
DECLARE TTEFFECT1B PIC X(80).	3827
DECLARE TTEFFECT2A PIC X(80).	3828
DECLARE TTEFFECT2B PIC X(80).	3829
DECLARE TTEFFECT3A PIC X(80).	3830
DECLARE TTEFFECT3B PIC X(80).	3831
DECLARE TTEFFECT4A PIC X(80).	3832
DECLARE TTEFFECT4B PIC X(80).	3833
DECLARE TTEFFECT5A PIC X(80).	3834
DECLARE TTEFFECT5B PIC X(80).	3835
DECLARE TTEFFECT6A PIC X(80).	3836
DECLARE TTEFFECT6B PIC X(80).	3837
DECLARE TTEFFECT1 PIC X(161).	3838
DECLARE TTEFFECT2 PIC X(161).	3839
DECLARE TTEFFECT3 PIC X(161).	3840
DECLARE TTEFFECT4 PIC X(161).	3841
DECLARE TTEFFECT5 PIC X(161).	3842
DECLARE TTEFFECT6 PIC X(161).	3843
DECLARE TTEFFECT PIC X(161).	3844
! =====	3845
! =====	3846
! =====	3847
! VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR DESCRIPTION AND	3848
! EFFECTS	3849
! =====	3850
! =====	3851
! =====	3852
! =====	3853
DECLARE TEMP1 PIC X(80).	3854

Datatrive Procedure FM_MODIFY (cont.)

DECLARE TEMP2	PIC X(80).	3855
DECLARE TEMP3	PIC X(80).	3856
DECLARE TEMP4	PIC X(80).	3857
DECLARE TEMP5	PIC X(242).	3858
DECLARE TEMP6	PIC X(984).	3859
!		3860
!		3861
=====		3862
!		3863
! VARIABLES USED AS FLAGS OR CONDITION INDICATORS		3864
!		3865
=====		3866
!		3867
DECLARE TCONTINUE1	PIC X(1).	3868
DECLARE TCONTINUE2	PIC X(1).	3869
DECLARE IMSG	PIC 9(2).	3870
DECLARE FLG1	PIC X(1).	3871
DECLARE FLG2	PIC X(1).	3872
DECLARE FLG3	PIC X(1).	3873
DECLARE FLG4	PIC X(1).	3874
!		3875
!		3876
=====		3877
!		3878
! VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM		3879
!		3880
=====		3881
!		3882
DECLARE TMSG	PIC X(80).	3883
DECLARE TMSG1	PIC X(80).	3884
DECLARE TMSG2	PIC X(80).	3885
DECLARE TMSG3	PIC X(80).	3886
DECLARE TMSG4	PIC X(80).	3887
DECLARE TMSG5	PIC X(80).	3888
DECLARE TMSG6	PIC X(80).	3889
DECLARE TMSG7	PIC X(80).	3890
DECLARE TMSG8	PIC X(80).	3891
DECLARE TMSG9	PIC X(80).	3892
DECLARE TMSG10	PIC X(80).	3893
DECLARE TMSG10A	PIC X(80).	3894
DECLARE TMSG11	PIC X(80).	3895
DECLARE TMSG12	PIC X(80).	3896
DECLARE TMSG13	PIC X(80).	3897
DECLARE TMSG14	PIC X(80).	3898
DECLARE TMSG15	PIC X(80).	3899
DECLARE TMSG16	PIC X(80).	3900
DECLARE TMSG17	PIC X(80).	3901
DECLARE TMSG18	PIC X(80).	3902
DECLARE TMSG19	PIC X(80).	3903
DECLARE TMSG20	PIC X(80).	3904
DECLARE TMSG21	PIC X(80).	3905


```

DECLARE TMSG22      PIC X(80).
DECLARE TMSG23      PIC X(80).
DECLARE TMSG24      PIC X(80).
!
!=====
!
!  VARIABLES USED AS COUNTERS
!=====
!
DECLARE ICNT        PIC 9(4).
DECLARE JCNT        PIC 9(4).
DECLARE KCNT        PIC 9(4).
DECLARE I           PIC 9(4).
DECLARE J           PIC 9(4).
DECLARE K           PIC 9(4).
!
!=====
!
!  VARIABLES USED TO TEMPORARILY STORE FAILURE MODE DATA FOR COMPARISON
!  OF INITIAL AND MODIFIED VALUES
!=====
!
DECLARE HDCREATED    USAGE DATE
                     EDIT_STRING X(23).
DECLARE HFMCODE      PIC X(20).
DECLARE HDESC        PIC X(242).
DECLARE HEFFECT1     PIC X(161).
DECLARE HEFFECT2     PIC X(161).
DECLARE HEFFECT3     PIC X(161).
DECLARE HEFFECT4     PIC X(161).
DECLARE HEFFECT5     PIC X(161).
DECLARE HEFFECT6     PIC X(161).
DECLARE HDLASTMOD    USAGE DATE
                     EDIT_STRING X(23).
DECLARE HMODPROC      PIC X(20).
!
!=====
!
!  VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE
!=====
!
DECLARE CAL          USAGE DATE
                     EDIT_STRING X(23).
!
!=====

```

Datatrieve Procedure FM_MODIFY (cont.)

```

===== 3957
! 3958
! READY THE DOMAINS SYSTEMS, MODULES, FAILUREMODES AND 3959
! FAILUREMODES_FORM 3960
! 3961
! ===== 3962
! 3963
SET ABORT 3964
READY FAILUREMODES SHARED WRITE 3965
READY SYSTEMS SHARED READ 3966
READY MODULES SHARED READ 3967
READY FAILUREMODES_FORM SHARED READ 3968
! 3969
! 3970
! ===== 3971
! 3972
! INITIALIZE THE MESSAGE VARIABLES AND INITIALIZE THE COUNTER (KCNT) 3973
! USED FOR NUMBERING THE LOG FILE RECORDS 3974
! 3975
! ===== 3976
! 3977
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 3978
TMSG2 = "DATA MUST BE ENTERED IN AT LEAST ONE OF THE FIELDS OR CONTINUE " | 3979
      "MUST BE N" 3980
TMSG3 = "SOURCE SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS" 3981
TMSG4 = "SOURCE MODULE IS NOT VALID -- MUST BE NUMERIC IN RANGE 1 TO " | 3982
      "9999 INCLUSIVE" 3983
TMSG5 = "SOURCE SYSTEM & MODULE IS NOT VALID -- NOT IN DOMAIN MODULES" 3984
TMSG6 = "FAILURE MODE & SUBMODE IS NOT VALID -- NOT IN TABLE " | 3985
      "FAILURE_MODE SUBMODE TABLE" 3986
TMSG7 = "ACCOMPLICE SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS" 3987
TMSG8 = "ACCOMPLICE MODULE IS NOT VALID -- MUST BE NUMERIC IN RANGE 0 " | 3988
      "TO 9999 INCLUSIVE" 3989
TMSG9 = "ACCOMPLICE SYSTEM & MODULE IS NOT VALID -- NOT IN DOMAIN MODULES" 3990
TMSG10 = "FAILURE" 3991
TMSG10A = "IS NOT VALID -- NOT IN DOMAIN FAILUREMODES" 3992
TMSG11 = "SOURCE MODULE IS NOT VALID -- NOT IN DOMAIN MODULES" 3993
TMSG12 = "ACCOMPLICE MODULE IS NOT VALID -- NOT IN DOMAIN MODULES" 3994
TMSG13 = "THE FIRST LINE OF THE FAILURE DESCRIPTION SHOULD NOT END WITH " | 3995
      "A HYPENATED WORD" 3996
TMSG14 = "THE SECOND LINE OF THE FAILURE DESCRIPTION SHOULD NOT END WITH " | 3997
      "A HYPENATED WORD" 3998
TMSG15 = "THE FIRST LINE OF EFFECT 1 SHOULD NOT END WITH A HYPENATED WORD" 3999
TMSG16 = "THE FIRST LINE OF EFFECT 2 SHOULD NOT END WITH A HYPENATED WORD" 4000
TMSG17 = "THE FIRST LINE OF EFFECT 3 SHOULD NOT END WITH A HYPENATED WORD" 4001
TMSG18 = "THE FIRST LINE OF EFFECT 4 SHOULD NOT END WITH A HYPENATED WORD" 4002
TMSG19 = "THE FIRST LINE OF EFFECT 5 SHOULD NOT END WITH A HYPENATED WORD" 4003
TMSG20 = "THE FIRST LINE OF EFFECT 6 SHOULD NOT END WITH A HYPENATED WORD" 4004
TMSG21 = "CONTINUE IS NOT VALID -- MUST BE Y OR N" 4005
TMSG22 = "NO RECORDS HAVE BEEN FOUND WITH THE DATA INDICATED ABOVE" 4006
TMSG23 = "ENTER MODIFICATIONS IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 4007

```

Datatrieve Procedure FM_MODIFY (cont.)

```

TMSG24 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"      4008
KCNT    = 0                                                  4009
!                                                            4010
!                                                            4011
=====                                                    4012
!                                                            4013
! PRIMARY LOOP TO MODIFY FAILURE MODES                        4014
!                                                            4015
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:             4016
! 1. LOOP TO REQUEST SEARCH DATA, TEST VALUES, PROMPT FOR CORRECTION 4017
!   OF INVALID INFORMATION AND COUNT THE RECORDS WHICH MATCH THE
!   SPECIFIED INPUT FIELDS                                     4018
! 2. IF TCONTINUE1 NE "N", SECTION TO DISPLAY THE MATCHING RECORDS 4019
!   ONE AT A TIME FOR POSSIBLE MODIFICATION (THIS SECTION IS
!   TERMINATED WHEN TCONTINUE2 = "N")                         4020
! 3. IF TCONTINUE1 = "N", SECTION TO REQUEST RESPONSE TO CONTINUE 4021
!   PROCEDURE OR EXIT TO MENU                                 4022
!                                                            4023
=====                                                    4024
!                                                            4025
!                                                            4026
TCONTINUE1 = "Y"                                             4027
WHILE TCONTINUE1 = "Y"                                       4028
BEGIN                                                         4029
!                                                            4030
!                                                            4031
=====                                                    4032
!                                                            4033
! LOOP TO DISPLAY BLANK TDMS FORMS, RETRIEVE THE DATA ENTERED ON THE 4034
! FORMS, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 4035
!                                                            4036
=====                                                    4037
!                                                            4038
!                                                            4039
FLG1 = "N"                                                    4040
IMSG = 1                                                       4041
WHILE FLG1 = "N"                                              4042
BEGIN                                                         4043
    IF IMSG = 1 THEN TMSG = TMSG1                             4044
    IF IMSG = 2 THEN TMSG = TMSG2                             4045
    IF IMSG = 3 THEN TMSG = TMSG3                             4046
    IF IMSG = 4 THEN TMSG = TMSG4                             4047
    IF IMSG = 5 THEN TMSG = TMSG5                             4048
    IF IMSG = 6 THEN TMSG = TMSG6                             4049
    IF IMSG = 7 THEN TMSG = TMSG7                             4050
    IF IMSG = 8 THEN TMSG = TMSG8                             4051
    IF IMSG = 9 THEN TMSG = TMSG9                             4052
    IF IMSG = 10 THEN TMSG = TMSG10 || " " | TSSYSTEM | " " | TSMODULE | 4053
                                         " " | TFMSUBM | " " | TAMODULE | 4054
                                         " " | TASYSTEM | " " | TAMODULE | 4055
                                         " " | TMSG10A | 4056
    IF IMSG = 11 THEN TMSG = TMSG11                           4057
    IF IMSG = 12 THEN TMSG = TMSG12                           4058

```

Datatrieve Procedure FM_MODIFY (cont.)

```

IF IMMSG = 13 THEN TMSG = TMSG13      4059
IF IMMSG = 14 THEN TMSG = TMSG14      4060
IF IMMSG = 15 THEN TMSG = TMSG15      4061
IF IMMSG = 16 THEN TMSG = TMSG16      4062
IF IMMSG = 17 THEN TMSG = TMSG17      4063
IF IMMSG = 18 THEN TMSG = TMSG18      4064
IF IMMSG = 19 THEN TMSG = TMSG19      4065
IF IMMSG = 20 THEN TMSG = TMSG20      4066
IF IMMSG = 21 THEN TMSG = TMSG21      4067
IF IMMSG = 22 THEN TMSG = TMSG22      4068
                                        4069
                                        4070
=====                               4071
THIS SECTION DISPLAYS THE FIND FAILURE MODES FORM AND
RETRIEVES THE DATA ENTERED ON THE FORM  4072
=====                               4073
                                        4074
                                        4075
                                        4076
FOR FIRST 1 FAILUREMODES_FORM           4077
  BEGIN                                4078
    DISPLAY FORM FAILUREMODES FIN1 FORM IN  4079
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING  4080
    BEGIN                                4081
      IF IMMSG NE 1 THEN                 4082
      BEGIN                               4083
        PUT FORM SOURCE_SYSTEM = TSSYSTEM  4084
        PUT FORM SOURCE_MODULE = TSMODULE  4085
        PUT FORM MODE SUBMODE = TFMSUBM    4086
        PUT FORM ACCOM_SYSTEM = TASYSTEM    4087
        PUT FORM ACCOM_MODULE = TAMODULE    4088
        PUT FORM DESCRIPTION_1 = TDESC1     4089
        PUT FORM DESCRIPTION_2 = TDESC2     4090
        PUT FORM DESCRIPTION_3 = TDESC3     4091
        PUT FORM EFFECT_1A = TEFFECT1A     4092
        PUT FORM EFFECT_1B = TEFFECT1B     4093
      END                                  4094
      PUT FORM MESSAGE = TMSG              4095
    END RETRIEVE USING                    4096
    BEGIN                                4097
      TSSYSTEM = GET FORM SOURCE_SYSTEM    4098
      TSMODULE = GET FORM SOURCE_MODULE    4099
      TFMSUBM = GET FORM MODE SUBMODE      4100
      TASYSTEM = GET FORM ACCOM_SYSTEM     4101
      TAMODULE = GET FORM ACCOM_MODULE     4102
      TDESC1 = GET FORM DESCRIPTION_1      4103
      TDESC2 = GET FORM DESCRIPTION_2      4104
      TDESC3 = GET FORM DESCRIPTION_3      4105
      TEFFECT1A = GET FORM EFFECT_1A       4106
      TEFFECT1B = GET FORM EFFECT_1B       4107
    END                                  4108
  END                                    4109

```

DISPLAY FORM FAILURE MODES FIN2 FORM IN	4110
DEV\$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING	4111
BEGIN	4112
IF MSG NE 1 THEN	4113
BEGIN	4114
PUT_FORM EFFECT_2A = TEFFECT2A	4115
PUT_FORM EFFECT_2B = TEFFECT2B	4116
PUT_FORM EFFECT_3A = TEFFECT3A	4117
PUT_FORM EFFECT_3B = TEFFECT3B	4118
PUT_FORM EFFECT_4A = TEFFECT4A	4119
PUT_FORM EFFECT_4B = TEFFECT4B	4120
PUT_FORM EFFECT_5A = TEFFECT5A	4121
PUT_FORM EFFECT_5B = TEFFECT5B	4122
PUT_FORM EFFECT_6A = TEFFECT6A	4123
PUT_FORM EFFECT_6B = TEFFECT6B	4124
END	4125
PUT_FORM MESSAGE = TMSG	4126
PUT_FORM CONTINUE = TCONTINUE1	4127
END RETRIEVE USING	4128
BEGIN	4129
TEFFECT2A = GET_FORM EFFECT_2A	4130
TEFFECT2B = GET_FORM EFFECT_2B	4131
TEFFECT3A = GET_FORM EFFECT_3A	4132
TEFFECT3B = GET_FORM EFFECT_3B	4133
TEFFECT4A = GET_FORM EFFECT_4A	4134
TEFFECT4B = GET_FORM EFFECT_4B	4135
TEFFECT5A = GET_FORM EFFECT_5A	4136
TEFFECT5B = GET_FORM EFFECT_5B	4137
TEFFECT6A = GET_FORM EFFECT_6A	4138
TEFFECT6B = GET_FORM EFFECT_6B	4139
TCONTINUE1 = GET_FORM CONTINUE	4140
END	4141
END	4142
TSMODULE1 = FN\$STR_EXTRACT(TSMODULE, 1, 1)	4143
TSMODULE2 = FN\$STR_EXTRACT(TSMODULE, 2, 1)	4144
TSMODULE3 = FN\$STR_EXTRACT(TSMODULE, 3, 1)	4145
TSMODULE4 = FN\$STR_EXTRACT(TSMODULE, 4, 1)	4146
TAMODULE1 = FN\$STR_EXTRACT(TAMODULE, 1, 1)	4147
TAMODULE2 = FN\$STR_EXTRACT(TAMODULE, 2, 1)	4148
TAMODULE3 = FN\$STR_EXTRACT(TAMODULE, 3, 1)	4149
TAMODULE4 = FN\$STR_EXTRACT(TAMODULE, 4, 1)	4150
	4151
	4152
=====	4153
IF TCONTINUE1 IS NOT EQUAL TO "N", ANY LEADING BLANKS WHICH	4154
WERE INADVERTENTLY ENTERED IN TDESC1, TDESC2, TDESC3,	4155
TEFFECT1A, TEFFECT1B, TEFFECT2A, TEFFECT2B, TEFFECT3A,	4156
TEFFECT3B, TEFFECT4A, TEFFECT4B, TEFFECT5A, TEFFECT5B,	4157
TEFFECT6A AND TEFFECT6B ARE REMOVED	4158
	4159
	4160

Datatrieve Procedure FM_MODIFY (cont.)

```

!      ===== 4161
! 4162
IF TCONTINUE1 NE "N" THEN 4163
  BEGIN 4164
    I = 1 4165
    WHILE I LE 15 4166
      BEGIN 4167
        IF I = 1 THEN TEMP1 = TDESC1 4168
        IF I = 2 THEN TEMP1 = TDESC2 4169
        IF I = 3 THEN TEMP1 = TDESC3 4170
        IF I = 4 THEN TEMP1 = TEFFECT1A 4171
        IF I = 5 THEN TEMP1 = TEFFECT1B 4172
        IF I = 6 THEN TEMP1 = TEFFECT2A 4173
        IF I = 7 THEN TEMP1 = TEFFECT2B 4174
        IF I = 8 THEN TEMP1 = TEFFECT3A 4175
        IF I = 9 THEN TEMP1 = TEFFECT3B 4176
        IF I = 10 THEN TEMP1 = TEFFECT4A 4177
        IF I = 11 THEN TEMP1 = TEFFECT4B 4178
        IF I = 12 THEN TEMP1 = TEFFECT5A 4179
        IF I = 13 THEN TEMP1 = TEFFECT5B 4180
        IF I = 14 THEN TEMP1 = TEFFECT6A 4181
        IF I = 15 THEN TEMP1 = TEFFECT6B 4182
        IF TEMP1 NE " " THEN 4183
          BEGIN 4184
            J = 1 4185
            WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " " 4186
              BEGIN 4187
                J = J + 1 4188
              END 4189
            TEMP1 = FN$STR_EXTRACT(TEMP1, J, 80 - J + 1) 4190
          END 4191
        IF I = 1 THEN TDESC1 = TEMP1 4192
        IF I = 2 THEN TDESC2 = TEMP1 4193
        IF I = 3 THEN TDESC3 = TEMP1 4194
        IF I = 4 THEN TEFFECT1A = TEMP1 4195
        IF I = 5 THEN TEFFECT1B = TEMP1 4196
        IF I = 6 THEN TEFFECT2A = TEMP1 4197
        IF I = 7 THEN TEFFECT2B = TEMP1 4198
        IF I = 8 THEN TEFFECT3A = TEMP1 4199
        IF I = 9 THEN TEFFECT3B = TEMP1 4200
        IF I = 10 THEN TEFFECT4A = TEMP1 4201
        IF I = 11 THEN TEFFECT4B = TEMP1 4202
        IF I = 12 THEN TEFFECT5A = TEMP1 4203
        IF I = 13 THEN TEFFECT5B = TEMP1 4204
        IF I = 14 THEN TEFFECT6A = TEMP1 4205
        IF I = 15 THEN TEFFECT6B = TEMP1 4206
        I = I + 1 4207
      END 4208
    END 4209
  END 4210
! 4211

```

Datatrieve Procedure FM_MODIFY (cont.)

```

=====
IF TCONTINUE1 IS NOT EQUAL TO "N", TESTS ARE PERFORMED TO
VERIFY ANY DATA ENTERED ON THE FORM. AS APPROPRIATE, THE
PROCEDURE CHECKS ANY OR ALL OF THE FOLLOWING:
1. DATA HAS BEEN ENTERED IN AT LEAST ONE OF THE FIELDS
2. TSSYSTEM IS IN DOMAIN SYSTEMS
3. TSMODULE IS NUMERIC IN RANGE 1 TO 9999 INCLUSIVE
4. TSSYSTEM | TSMODULE IS IN DOMAIN MODULES
5. TFMSUBM IS IN TABLE FAILURE_MODE_SUBMODE_TABLE
6. TASYSTEM IS IN DOMAIN SYSTEMS
7. TAMODULE IS NUMERIC IN RANGE 0 TO 9999 INCLUSIVE
8. TASYSTEM | TAMODULE IS IN DOMAIN MODULES
9. TSSYSTEM | TSMODULE | TFMSUBM | TASYSTEM | TAMODULE
   IS IN DOMAIN FAILUREMODES
10. TSMODULE IS IN DOMAIN MODULES
11. TAMODULE IS IN DOMAIN MODULES
12. TDESC1 DOES NOT END WITH A HYPHENATED WORD
13. TDESC2 DOES NOT END WITH A HYPHENATED WORD
14. TEFFECT1A DOES NOT END WITH A HYPHENATED WORD
15. TEFFECT2A DOES NOT END WITH A HYPHENATED WORD
16. TEFFECT3A DOES NOT END WITH A HYPHENATED WORD
17. TEFFECT4A DOES NOT END WITH A HYPHENATED WORD
18. TEFFECT5A DOES NOT END WITH A HYPHENATED WORD
19. TEFFECT6A DOES NOT END WITH A HYPHENATED WORD
20. TCONTINUE1 IS "Y" OR "N"
21. AT LEAST ONE RECORD EXISTS WITH THE DATA SPECIFIED
=====

FLG1 = "Y"
IF TCONTINUE1 NE "N" THEN
  BEGIN
    IF TSSYSTEM = " " AND TSMODULE = " " AND
       TFMSUBM = " " AND
       TASYSTEM = " " AND TAMODULE = " " AND
       TDESC1 = " " AND TDESC2 = " " AND TDESC3 = " " AND
       TEFFECT1A = " " AND TEFFECT1B = " " AND
       TEFFECT2A = " " AND TEFFECT2B = " " AND
       TEFFECT3A = " " AND TEFFECT3B = " " AND
       TEFFECT4A = " " AND TEFFECT4B = " " AND
       TEFFECT5A = " " AND TEFFECT5B = " " AND
       TEFFECT6A = " " AND TEFFECT6B = " " THEN
      BEGIN
        FLG1 = "N"
        MSG = 2
      END
    IF FLG1 = "Y" AND
       TSSYSTEM NE " " THEN
      BEGIN
        FLG2 = "N"

```

Datatrieve Procedure FM_MODIFY (cont.)

```

FOR FIRST 1 SYSTEMS WITH                                4263
    SYSTEM = TSSYSTEM  FLG2 = "Y"                      4264
IF FLG2 = "N" THEN                                     4265
    BEGIN                                              4266
        FLG1 = "N"                                     4267
        IMSG = 3                                       4268
    END                                              4269
END                                                  4270
IF FLG1      = "Y" AND                                4271
    TSMODULE NE " " THEN                              4272
    BEGIN                                              4273
        FLG2 = "N"                                     4274
        IF (TSMODULE1 = " " OR TSMODULE1 IN NUMBER_TABLE) AND 4275
            (TSMODULE2 = " " OR TSMODULE2 IN NUMBER_TABLE) AND 4276
            (TSMODULE3 = " " OR TSMODULE3 IN NUMBER_TABLE) AND 4277
            (TSMODULE4 = " " OR TSMODULE4 IN NUMBER_TABLE) THEN 4278
            BEGIN                                      4279
                NSMODULE = TSMODULE                  4280
                IF NSMODULE GT 0 THEN                  4281
                    FLG2 = "Y"                        4282
                END                                    4283
            END IF FLG2 = "N"                          4284
        BEGIN                                          4285
            FLG1 = "N"                                 4286
            IMSG = 4                                   4287
        END                                          4288
    END                                          4289
IF FLG1      = "Y" AND                                4290
    TSSYSTEM NE " " AND                              4291
    TSMODULE NE " " THEN                              4292
    BEGIN                                              4293
        FLG2 = "N"                                     4294
        TSMODULE = NSMODULE                           4295
        IF NSMODULE LT 1000 THEN TSMODULE = "0" | TSMODULE 4296
        IF NSMODULE LT 100  THEN TSMODULE = "0" | TSMODULE 4297
        IF NSMODULE LT 10   THEN TSMODULE = "0" | TSMODULE 4298
        FOR FIRST 1 MODULES WITH                      4299
            SYSTEM MODULE = TSSYSTEM | TSMODULE  FLG2 = "Y" 4300
        IF FLG2 = "N" THEN                              4301
            BEGIN                                      4302
                FLG1 = "N"                             4303
                IMSG = 5                                 4304
            END                                      4305
        END                                          4306
IF FLG1      = "Y" AND                                4307
    TFMSUBM NE " " THEN                              4308
    BEGIN                                              4309
        IF TFMSUBM NOT IN FAILURE_MODE_SUBMODE_TABLE THEN 4310
            BEGIN                                      4311
                FLG1 = "N"                             4312
                IMSG = 6                                 4313
            END
        END
    END

```


Datatrieve Procedure FM_MODIFY (cont.)

```

      END                                     4314
    END                                     4315
  IF FLG1 = "Y" AND                         4316
    TASYSTEM NE " " THEN                   4317
    BEGIN                                  4318
      FLG2 = "N"                           4319
      IF TASYSTEM = "----" THEN FLG2 = "Y" 4320
      IF FLG2 = "N" THEN                   4321
        FOR FIRST 1 SYSTEMS WITH           4322
          SYSTEM = TASYSTEM FLG2 = "Y"     4323
      IF FLG2 = "N" THEN                   4324
        BEGIN                              4325
          FLG1 = "N"                       4326
          IMMSG = 7                        4327
        END                                4328
      END                                  4329
    IF FLG1 = "Y" AND                       4330
      TAMODULE NE " " THEN                 4331
      BEGIN                                4332
        FLG2 = "N"                         4333
        IF (TAMODULE1 = " " OR TAMODULE1 IN NUMBER_TABLE) AND 4334
          (TAMODULE2 = " " OR TAMODULE2 IN NUMBER_TABLE) AND 4335
          (TAMODULE3 = " " OR TAMODULE3 IN NUMBER_TABLE) AND 4336
          (TAMODULE4 = " " OR TAMODULE4 IN NUMBER_TABLE) THEN 4337
          BEGIN                            4338
            NAMODULE = TAMODULE            4339
            FLG2 = "Y"                     4340
          END                              4341
        IF FLG2 = "N"                       4342
          BEGIN                             4343
            FLG1 = "N"                     4344
            IMMSG = 8                      4345
          END                              4346
        END                                4347
      END                                  4348
    IF FLG1 = "Y" AND                       4349
      TASYSTEM NE " " AND                  4350
      TAMODULE NE " " THEN                 4351
      BEGIN                                4352
        FLG2 = "N"                         4353
        TAMODULE = NAMODULE                 4354
        IF NAMODULE LT 1000 THEN TAMODULE = "0" | TAMODULE 4355
        IF NAMODULE LT 100 THEN TAMODULE = "0" | TAMODULE 4356
        IF NAMODULE LT 10 THEN TAMODULE = "0" | TAMODULE 4357
        IF TASYSTEM | TAMODULE = "----0000" THEN FLG2 = "Y" 4358
        IF FLG2 = "N" THEN                   4359
          FOR FIRST 1 MODULES WITH           4360
            SYSTEM_MODULE = TASYSTEM | TAMODULE FLG2 = "Y" 4361
        IF FLG2 = "N" THEN                   4362
          BEGIN                              4363
            FLG1 = "N"                       4364
            IMMSG = 9

```

Datatrieve Procedure FM_MODIFY (cont.)

```

      END
    END
  IF FLG1 = "Y" AND
    TSSYSTEM NE " " AND TSMODULE NE " " AND
    TFMSUBM NE " " AND
    TASYSTEM NE " " AND TAMODULE NE " " THEN
    BEGIN
      FLG2 = "N"
      FOR FAILUREMODES WITH
        FMCODE = TSSYSTEM | TSMODULE | TFMSUBM |
                  TASYSTEM | TAMODULE FLG2 = "Y"
      IF FLG2 = "N" THEN
        BEGIN
          FLG1 = "N"
          IMSG = 10
        END
      END
    END
  IF FLG1 = "Y" AND
    TSSYSTEM = " " AND
    TSMODULE NE " " THEN
    BEGIN
      FLG2 = "N"
      FOR FIRST 1 MODULES WITH
        MODULE = NSMODULE FLG2 = "Y"
      IF FLG2 = "N" THEN
        BEGIN
          FLG1 = "N"
          IMSG = 11
        END
      END
    END
  IF FLG1 = "Y" AND
    TASYSTEM = " " AND
    TAMODULE NE " " THEN
    BEGIN
      FLG2 = "N"
      TAMODULE = NAMODULE
      IF NAMODULE LT 1000 THEN TAMODULE = "0" | TAMODULE
      IF NAMODULE LT 100 THEN TAMODULE = "0" | TAMODULE
      IF NAMODULE LT 10 THEN TAMODULE = "0" | TAMODULE
      IF TAMODULE = "0000" THEN FLG2 = "Y"
      IF FLG2 = "N" THEN
        FOR FIRST 1 MODULES WITH
          MODULE = NAMODULE FLG2 = "Y"
      IF FLG2 = "N" THEN
        BEGIN
          FLG1 = "N"
          IMSG = 12
        END
      END
    END
  IF FLG1 = "Y" THEN
    BEGIN

```

Datatrieve Procedure FM_MODIFY (cont.)

```

I = 1                                         4416
WHILE FLG1 = "Y" AND I LE 8                 4417
  BEGIN                                     4418
    IF I = 1 THEN TEMP1 = TDESC1            4419
    IF I = 2 THEN TEMP1 = TDESC2            4420
    IF I = 3 THEN TEMP1 = TEFFECT1A         4421
    IF I = 4 THEN TEMP1 = TEFFECT2A         4422
    IF I = 5 THEN TEMP1 = TEFFECT3A         4423
    IF I = 6 THEN TEMP1 = TEFFECT4A         4424
    IF I = 7 THEN TEMP1 = TEFFECT5A         4425
    IF I = 8 THEN TEMP1 = TEFFECT6A         4426
    IF TEMP1 NE " " THEN                    4427
      BEGIN                                 4428
        J = 80                             4429
        WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " " 4430
          BEGIN                             4431
            J = J - 1                       4432
          END                               4433
          IF FN$STR_EXTRACT(TEMP1,           4434
            J, 1) = "-" AND                 4435
            FN$STR_EXTRACT(TEMP1,           4436
            J - 1, 1) NE " " AND            4437
            FN$STR_EXTRACT(TEMP1,           4438
            J - 2, 2) NE " -" THEN          4439
            BEGIN                           4440
              FLG1 = "N"                    4441
              IF I = 1 THEN MSG = 13         4442
              IF I = 2 THEN MSG = 14         4443
              IF I = 3 THEN MSG = 15         4444
              IF I = 4 THEN MSG = 16         4445
              IF I = 5 THEN MSG = 17         4446
              IF I = 6 THEN MSG = 18         4447
              IF I = 7 THEN MSG = 19         4448
              IF I = 8 THEN MSG = 20         4449
            END                             4450
          END                               4451
        I = I + 1                           4452
      END                                   4453
    END                                     4454
  IF FLG1 = "Y" THEN                         4455
    BEGIN                                    4456
      IF TCONTINUE1 NE "Y" AND              4457
      TCONTINUE1 NE "N" THEN                4458
        BEGIN                               4459
          FLG1 = "N"                        4460
          MSG = 21                          4461
        END                                 4462
      END                                   4463
    IF FLG1 = "Y" THEN                       4464
      BEGIN                                  4465
        TDESC = TDESC1                     4466

```

Datatrieve Procedure FM_MODIFY (cont.)

```

IF TDESC = " " THEN 4467
  TDESC = TDESC2 ELSE 4468
  TDESC = TDESC || " " | TDESC2 4469
IF TDESC = " " THEN 4470
  TDESC = TDESC3 ELSE 4471
  TDESC = TDESC || " " | TDESC3 4472
TEFFECT1 = TEFFECT1A 4473
IF TEFFECT1 = " " THEN 4474
  TEFFECT1 = TEFFECT1B ELSE 4475
  TEFFECT1 = TEFFECT1 || " " | TEFFECT1B 4476
TEFFECT2 = TEFFECT2A 4477
IF TEFFECT2 = " " THEN 4478
  TEFFECT2 = TEFFECT2B ELSE 4479
  TEFFECT2 = TEFFECT2 || " " | TEFFECT2B 4480
TEFFECT3 = TEFFECT3A 4481
IF TEFFECT3 = " " THEN 4482
  TEFFECT3 = TEFFECT3B ELSE 4483
  TEFFECT3 = TEFFECT3 || " " | TEFFECT3B 4484
TEFFECT4 = TEFFECT4A 4485
IF TEFFECT4 = " " THEN 4486
  TEFFECT4 = TEFFECT4B ELSE 4487
  TEFFECT4 = TEFFECT4 || " " | TEFFECT4B 4488
TEFFECT5 = TEFFECT5A 4489
IF TEFFECT5 = " " THEN 4490
  TEFFECT5 = TEFFECT5B ELSE 4491
  TEFFECT5 = TEFFECT5 || " " | TEFFECT5B 4492
TEFFECT6 = TEFFECT6A 4493
IF TEFFECT6 = " " THEN 4494
  TEFFECT6 = TEFFECT6B ELSE 4495
  TEFFECT6 = TEFFECT6 || " " | TEFFECT6B 4496
END 4497
IF FLG1 = "Y" THEN 4498
  BEGIN 4499
    FLG2 = "N" 4500
    IF TSSYSTEM NE " " AND TSMODULE NE " " AND 4501
      TFMSUBM NE " " AND 4502
      TASYSTEM NE " " AND TAMODULE NE " " THEN 4503
      BEGIN 4504
        FLG2 = "Y" 4505
        JCNT = 0 4506
        FOR FAILUREMODES WITH 4507
          FMCODE = TSSYSTEM | TSMODULE | TFMSUBM | 4508
            TASYSTEM | TAMODULE 4509
          BEGIN 4510
            FLG3 = "Y" 4511
            IF TDESC NE " " AND 4512
              DESCRIPTION NE TDESC THEN FLG3 = "N" 4513
            IF TEFFECT1 NE " " AND 4514
              EFFECT1 NE TEFFECT1 THEN FLG3 = "N" 4515
            IF TEFFECT2 NE " " AND 4516
              EFFECT2 NE TEFFECT2 THEN FLG3 = "N" 4517

```

Datatrieve Procedure FM_MODIFY (cont.)

```

IF TEFFECT3 NE " " AND 4518
EFFECT3 NE TEFFECT3 THEN FLG3 = "N" 4519
IF TEFFECT4 NE " " AND 4520
EFFECT4 NE TEFFECT4 THEN FLG3 = "N" 4521
IF TEFFECT5 NE " " AND 4522
EFFECT5 NE TEFFECT5 THEN FLG3 = "N" 4523
IF TEFFECT6 NE " " AND 4524
EFFECT6 NE TEFFECT6 THEN FLG3 = "N" 4525
IF FLG3 = "Y" THEN JCNT = JCNT + 1 4526
END 4527
IF JCNT = 0 THEN 4528
BEGIN 4529
FLG1 = "N" 4530
IMSG = 22 4531
END 4532
END 4533
IF FLG2 = "N" THEN 4534
BEGIN 4535
JCNT = 0 4536
FOR FAILUREMODES 4537
BEGIN 4538
FLG3 = "Y" 4539
IF TSSYSTEM NE " " AND 4540
SOURCE_SYSTEM NE TSSYSTEM THEN 4541
FLG3 = "N" 4542
IF TSMODULE NE " " AND 4543
SOURCE_MODULE NE NSMODULE THEN 4544
FLG3 = "N" 4545
IF TFMSUBM NE " " AND 4546
FAILURE_MODE_SUBMODE NE TFMSUBM THEN 4547
FLG3 = "N" 4548
IF TASYSTEM NE " " AND 4549
ACCOMPLICE_SYSTEM NE TASYSTEM THEN 4550
FLG3 = "N" 4551
IF TAMODULE NE " " AND 4552
ACCOMPLICE_MODULE NE NAMODULE THEN 4553
FLG3 = "N" 4554
IF TDESC NE " " AND 4555
DESCRIPTION NE TDESC THEN FLG3 = "N" 4556
IF TEFFECT1 NE " " AND 4557
EFFECT1 NE TEFFECT1 THEN FLG3 = "N" 4558
IF TEFFECT2 NE " " AND 4559
EFFECT2 NE TEFFECT2 THEN FLG3 = "N" 4560
IF TEFFECT3 NE " " AND 4561
EFFECT3 NE TEFFECT3 THEN FLG3 = "N" 4562
IF TEFFECT4 NE " " AND 4563
EFFECT4 NE TEFFECT4 THEN FLG3 = "N" 4564
IF TEFFECT5 NE " " AND 4565
EFFECT5 NE TEFFECT5 THEN FLG3 = "N" 4566
IF TEFFECT6 NE " " AND 4567
EFFECT6 NE TEFFECT6 THEN FLG3 = "N" 4568

```

Datatrieve Procedure FM_MODIFY (cont.)

```

                                IF FLG3 = "Y" THEN JCNT = JCNT + 1      4569
                                END                                    4570
                                IF JCNT = 0 THEN                      4571
                                BEGIN                                  4572
                                    FLG1 = "N"                        4573
                                    IMSG = 22                        4574
                                END                                    4575
                                END                                    4576
                                END                                    4577
                                END                                    4578
                                END                                    4579
                                !                                     4580
                                !                                     4581
                                !                                     4582
                                !                                     4583
                                !                                     4584
                                !                                     4585
                                !                                     4586
                                !                                     4587
                                !                                     4588
                                !                                     4589
                                !                                     4590
                                !                                     4591
                                !                                     4592
                                !                                     4593
                                !                                     4594
                                !                                     4595
                                !                                     4596
                                !                                     4597
                                !                                     4598
                                !                                     4599
                                !                                     4600
                                !                                     4601
                                !                                     4602
                                !                                     4603
                                !                                     4604
                                !                                     4605
                                !                                     4606
                                !                                     4607
                                !                                     4608
                                !                                     4609
                                !                                     4610
                                !                                     4611
                                !                                     4612
                                !                                     4613
                                !                                     4614
                                !                                     4615
                                !                                     4616
                                !                                     4617
                                !                                     4618
                                !                                     4619

```

IF TCONTINUE1 IS NOT EQUAL TO "N", THIS SECTION DISPLAYS THE
 INDICATED RECORDS ONE AT A TIME AND ALLOWS CORRECTIONS TO BE MADE
 TO ANY OR ALL OF THE RECORDS. THE FIELDS SOURCE_SYSTEM,
 SOURCE_MODULE, FAILURE_MODE_SUBMODE, ACCOMPLICE_SYSTEM AND
 ACCOMPLICE_MODULE CANNOT BE CHANGED SINCE THESE FIVE FIELDS
 UNIQUELY DEFINE THE FAILURE MODE.

```

IF TCONTINUE1 NE "N" THEN
    BEGIN
        TCONTINUE2 = "Y"
        FLG2 = "N"
        IF TSSYSTEM NE " " AND TSMODULE NE " " AND
           TFMSUBM NE " " AND
           TASYSTEM NE " " AND TAMODULE NE " " THEN
            BEGIN
                FLG2 = "Y"
                ICNT = 0
                FOR FAILUREMODES WITH
                   FMCODE = TSSYSTEM | TSMODULE | TFMSUBM |
                           TASYSTEM | TAMODULE | SORTED BY
                           FMCODE
                BEGIN
                    IF TCONTINUE2 = "Y" THEN
                        BEGIN
                            FLG3 = "Y"
                            IF TDESC NE " " AND
                               DESCRIPTION NE TDESC THEN FLG3 = "N"
                            IF TEFFECT1 NE " " AND
                               EFFECT1 NE TEFFECT1 THEN FLG3 = "N"
                            IF TEFFECT2 NE " " AND
                               EFFECT2 NE TEFFECT2 THEN FLG3 = "N"
                            IF TEFFECT3 NE " " AND
                               EFFECT3 NE TEFFECT3 THEN FLG3 = "N"
                            IF TEFFECT4 NE " " AND

```

Datatrieve Procedure FM_MODIFY (cont.)

```

        EFFECT4 NE TEFFECT4 THEN FLG3 = "N" 4620
    IF TEFFECT5 NE " " AND 4621
        EFFECT5 NE TEFFECT5 THEN FLG3 = "N" 4622
    IF TEFFECT6 NE " " AND 4623
        EFFECT6 NE TEFFECT6 THEN FLG3 = "N" 4624
    IF FLG3 = "Y" THEN 4625
        BEGIN 4626
            ICNT = ICNT + 1 4627
            :FM_MODIFY_1 4628
        END 4629
    END 4630
END 4631
END 4632
IF FLG2 = "N" THEN 4633
    BEGIN 4634
        ICNT = 0 4635
        FOR FAILUREMODES SORTED BY FMCODE 4636
            BEGIN 4637
                IF TCONTINUE2 = "Y" THEN 4638
                    BEGIN 4639
                        FLG3 = "Y" 4640
                        IF TSSYSTEM NE " " AND 4641
                            SOURCE_SYSTEM NE TSSYSTEM THEN 4642
                            FLG3 = "N" 4643
                        IF TSMODULE NE " " AND 4644
                            SOURCE_MODULE NE NSMODULE THEN 4645
                            FLG3 = "N" 4646
                        IF TFMSUBM NE " " AND 4647
                            FAILURE_MODE_SUBMODE NE TFMSUBM THEN 4648
                            FLG3 = "N" 4649
                        IF TASYSTEM NE " " AND 4650
                            ACCOMPLICE_SYSTEM NE TASYSTEM THEN 4651
                            FLG3 = "N" 4652
                        IF TAMODULE NE " " AND 4653
                            ACCOMPLICE_MODULE NE NAMODULE THEN 4654
                            FLG3 = "N" 4655
                        IF TDESC NE " " AND 4656
                            DESCRIPTION NE TDESC THEN FLG3 = "N" 4657
                        IF TEFFECT1 NE " " AND 4658
                            EFFECT1 NE TEFFECT1 THEN FLG3 = "N" 4659
                        IF TEFFECT2 NE " " AND 4660
                            EFFECT2 NE TEFFECT2 THEN FLG3 = "N" 4661
                        IF TEFFECT3 NE " " AND 4662
                            EFFECT3 NE TEFFECT3 THEN FLG3 = "N" 4663
                        IF TEFFECT4 NE " " AND 4664
                            EFFECT4 NE TEFFECT4 THEN FLG3 = "N" 4665
                        IF TEFFECT5 NE " " AND 4666
                            EFFECT5 NE TEFFECT5 THEN FLG3 = "N" 4667
                        IF TEFFECT6 NE " " AND 4668
                            EFFECT6 NE TEFFECT6 THEN FLG3 = "N" 4669
                    IF FLG3 = "Y" THEN 4670

```

Datatrieve Procedure FM_MODIFY (cont.)

```

                                BEGIN                                4671
                                ICNT = ICNT + 1                      4672
                                :FM_MODIFY_1                        4673
                                END                                  4674
                                END                                  4675
                                END                                  4676
                                END                                  4677
                                END                                  4678
                                END                                  4679
                                END                                  4680
                                END                                  4681
                                END                                  4682
                                END                                  4683
                                END                                  4684
                                END                                  4685
                                END                                  4686
                                END                                  4687
                                END                                  4688
                                END                                  4689
                                END                                  4690
                                END                                  4691
                                END                                  4692
                                END                                  4693
                                END                                  4694
                                END                                  4695
                                END                                  4696
                                END                                  4697
                                END                                  4698
                                END                                  4699
                                END                                  4700
                                END                                  4701
                                END                                  4702
                                END                                  4703
                                END                                  4704

```

=====

IF TCONTINUE1 IS EQUAL TO "N", A RESPONSE IS REQUESTED TO CONTINUE
MODIFYING FAILURE MODES

=====

```

IF TCONTINUE1 = "N" THEN
  BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    TCONTINUE1 = "X"
    WHILE TCONTINUE1 NE "Y" AND
      TCONTINUE1 NE "N"
      BEGIN
        PRINT SKIP 2,
          "Do you wish to continue modifying", SKIP 1,
          "FAILURE MODES?", SKIP 1
        TCONTINUE1 = FN$UPCASE(*."Y or N")
        PRINT " "
      END
    END
  END
END
END-PROCEDURE

```


Datatrieve Procedure FM_MODIFY_1

```

DEFINE PROCEDURE FM_MODIFY_1                                4705
!                                                            4706
!                                                            4707
!=====                                                    4708
!                                                            4709
! THE FIELDS OF THE INCOMING FAILURE MODE RECORD ARE ASSIGNED TO  4710
! VARIABLES FOR DISPLAY AND MODIFICATION.  THE FIELDS SOURCE_SYSTEM, 4711
! SOURCE_MODULE, FAILURE_MODE_SUBMODE, ACCOMPLICE_SYSTEM AND        4712
! ACCOMPLICE_MODULE ARE NOT ASSIGNED TO VARIABLES SINCE THESE FIELDS 4713
! CANNOT BE MODIFIED.                                             4714
!                                                            4715
!=====                                                    4716
!                                                            4717
TTDESC      = DESCRIPTION                                     4718
TTEFFECT1   = EFFECT1                                       4719
TTEFFECT2   = EFFECT2                                       4720
TTEFFECT3   = EFFECT3                                       4721
TTEFFECT4   = EFFECT4                                       4722
TTEFFECT5   = EFFECT5                                       4723
TTEFFECT6   = EFFECT6                                       4724
!                                                            4725
!=====                                                    4726
!                                                            4727
! THIS SECTION SPLITS THE COMPOSITE VARIABLES TTDESC AND TTEFFECT1 4728
! THROUGH TTEFFECT6 INTO VARIABLES WHICH ARE 80 CHARACTERS LONG FOR 4729
! DISPLAY ON THE TDMS FORM                                         4730
!=====                                                    4731
!                                                            4732
!=====                                                    4733
!                                                            4734
I = 1                                                       4735
WHILE I LE 7                                                4736
  BEGIN                                                    4737
    IF I = 1 THEN TEMP5 = TTDESC                            4738
    IF I = 2 THEN TEMP5 = TTEFFECT1                         4739
    IF I = 3 THEN TEMP5 = TTEFFECT2                         4740
    IF I = 4 THEN TEMP5 = TTEFFECT3                         4741
    IF I = 5 THEN TEMP5 = TTEFFECT4                         4742
    IF I = 6 THEN TEMP5 = TTEFFECT5                         4743
    IF I = 7 THEN TEMP5 = TTEFFECT6                         4744
    J      = 1                                              4745
    TEMP1 = " "                                             4746
    TEMP2 = " "                                             4747
    TEMP3 = " "                                             4748
    WHILE J LE 3                                           4749
      BEGIN                                              4750
        IF TEMP5 NE " " THEN                             4751
          BEGIN                                           4752
            K = 1                                          4753
            WHILE FN$STR_EXTRACT(TEMP5, K, 1) = " "      4754
              BEGIN                                       4755

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

        K = K + 1                                4756
    END                                            4757
    TEMP5 = FN$STR_EXTRACT(TEMP5, K, 242 - K + 1) 4758
END                                              4759
FLG4 = "N"                                       4760
IF FN$STR_EXTRACT(TEMP5, 80, 1) = " " OR        4761
   FN$STR_EXTRACT(TEMP5, 81, 1) = " " THEN      4762
    BEGIN                                       4763
        TEMP4 = FN$STR_EXTRACT(TEMP5, 1, 80)   4764
        TEMP5 = FN$STR_EXTRACT(TEMP5, 81, 162) 4765
        FLG4 = "Y"                             4766
    END                                         4767
IF FLG4 = "N" THEN                             4768
    BEGIN                                       4769
        K = 80                                4770
        WHILE FN$STR_EXTRACT(TEMP5, K, 1) NE " " AND 4771
            K GT 1                             4772
        BEGIN                                   4773
            K = K - 1                           4774
        END                                     4775
        CHOICE                                  4776
            K = 1 THEN TEMP4 = FN$STR_EXTRACT(TEMP5, 1, 80) 4777
            K GT 1 THEN TEMP4 = FN$STR_EXTRACT(TEMP5, 1, K - 1 + 1) 4778
        END_CHOICE                             4779
        CHOICE                                  4780
            K = 1 THEN TEMP5 = FN$STR_EXTRACT(TEMP5, 81, 162) 4781
            K GT 1 THEN TEMP5 = FN$STR_EXTRACT(TEMP5, K + 1, 4782
                242 - (K + 1) + 1)              4783
        END_CHOICE                             4784
    END                                         4785
    IF J = 1 THEN TEMP1 = TEMP4                 4786
    IF J = 2 THEN TEMP2 = TEMP4                 4787
    IF J = 3 THEN TEMP3 = TEMP4                 4788
    J = J + 1                                   4789
END                                              4790
IF I = 1 THEN                                  4791
    BEGIN                                       4792
        TTDESC1 = TEMP1                       4793
        TTDESC2 = TEMP2                       4794
        TTDESC3 = TEMP3                       4795
    END                                         4796
IF I = 2 THEN                                  4797
    BEGIN                                       4798
        TTEFFECT1A = TEMP1                    4799
        TTEFFECT1B = TEMP2                    4800
    END                                         4801
IF I = 3 THEN                                  4802
    BEGIN                                       4803
        TTEFFECT2A = TEMP1                    4804
        TTEFFECT2B = TEMP2                    4805
    END                                         4806

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

IF I = 4 THEN                                4807
  BEGIN                                      4808
    TTEFFECT3A = TEMP1                      4809
    TTEFFECT3B = TEMP2                      4810
  END                                        4811
IF I = 5 THEN                                4812
  BEGIN                                      4813
    TTEFFECT4A = TEMP1                      4814
    TTEFFECT4B = TEMP2                      4815
  END                                        4816
IF I = 6 THEN                                4817
  BEGIN                                      4818
    TTEFFECT5A = TEMP1                      4819
    TTEFFECT5B = TEMP2                      4820
  END                                        4821
IF I = 7 THEN                                4822
  BEGIN                                      4823
    TTEFFECT6A = TEMP1                      4824
    TTEFFECT6B = TEMP2                      4825
  END                                        4826
  I = I + 1                                4827
END                                          4828
!                                          4829
!                                          4830
!=====                                4831
!                                          4832
! LOOP TO DISPLAY A FAILURE MODE RECORD USING A TDMS FORM, RETRIEVE DATA  4833
! FROM THE FORM, TEST THE INCOMING INFORMATION AND REQUEST CORRECTION OF  4834
! INVALID DATA                                4835
!=====                                4836
!                                          4837
!                                          4838
FLG4 = "N"                                4839
IMSG = 23                                4840
WHILE FLG4 = "N"                            4841
  BEGIN                                    4842
    IF IMSG = 13 THEN TMSG = TMSG13        4843
    IF IMSG = 14 THEN TMSG = TMSG14        4844
    IF IMSG = 15 THEN TMSG = TMSG15        4845
    IF IMSG = 16 THEN TMSG = TMSG16        4846
    IF IMSG = 17 THEN TMSG = TMSG17        4847
    IF IMSG = 18 THEN TMSG = TMSG18        4848
    IF IMSG = 19 THEN TMSG = TMSG19        4849
    IF IMSG = 20 THEN TMSG = TMSG20        4850
    IF IMSG = 23 THEN TMSG = TMSG23        4851
    IF IMSG = 24 THEN TMSG = TMSG24        4852
!                                          4853
!                                          4854
!=====                                4855
! THIS SECTION DISPLAYS THE MODIFY FAILURE MODES FORM AND                4856
!                                          4857

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

!      RETRIEVES THE DATA ENTERED ON THE FORM                                4858
!      =====                                                                    4859
!      =====                                                                    4860
!      =====                                                                    4861
!
DISPLAY FORM FAILUREMODES MOD1 FORM IN                                         4862
DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING                                   4863
BEGIN                                                                           4864
    PUT FORM RECORD NUMBER = ICNT                                             4865
    PUT FORM TOTAL RECORDS = JCNT                                             4866
    PUT FORM SOURCE_SYSTEM = SOURCE_SYSTEM                                   4867
    PUT FORM SOURCE_MODULE = SOURCE_MODULE                                   4868
    PUT FORM MODE SUBMODE = FAILURE_MODE SUBMODE                             4869
    PUT FORM ACCOM_SYSTEM = ACCOMPLICE_SYSTEM                               4870
    PUT FORM ACCOM_MODULE = ACCOMPLICE_MODULE                               4871
    PUT FORM DESCRIPTION_1 = TTDESC1                                          4872
    PUT FORM DESCRIPTION_2 = TTDESC2                                          4873
    PUT FORM DESCRIPTION_3 = TTDESC3                                          4874
    PUT FORM EFFECT_1A = TTEFFECT1A                                           4875
    PUT FORM EFFECT_1B = TTEFFECT1B                                           4876
    PUT FORM MESSAGE = TMSG                                                  4877
END RETRIEVE USING                                                            4878
BEGIN                                                                           4879
    TTDESC1 = GET FORM DESCRIPTION_1                                           4880
    TTDESC2 = GET FORM DESCRIPTION_2                                           4881
    TTDESC3 = GET FORM DESCRIPTION_3                                           4882
    TTEFFECT1A = GET FORM EFFECT_1A                                           4883
    TTEFFECT1B = GET FORM EFFECT_1B                                           4884
END                                                                           4885
DISPLAY FORM FAILUREMODES MOD2 FORM IN                                         4886
DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING                                   4887
BEGIN                                                                           4888
    PUT FORM RECORD NUMBER = ICNT                                             4889
    PUT FORM TOTAL RECORDS = JCNT                                             4890
    PUT FORM EFFECT_2A = TTEFFECT2A                                           4891
    PUT FORM EFFECT_2B = TTEFFECT2B                                           4892
    PUT FORM EFFECT_3A = TTEFFECT3A                                           4893
    PUT FORM EFFECT_3B = TTEFFECT3B                                           4894
    PUT FORM EFFECT_4A = TTEFFECT4A                                           4895
    PUT FORM EFFECT_4B = TTEFFECT4B                                           4896
    PUT FORM EFFECT_5A = TTEFFECT5A                                           4897
    PUT FORM EFFECT_5B = TTEFFECT5B                                           4898
    PUT FORM EFFECT_6A = TTEFFECT6A                                           4899
    PUT FORM EFFECT_6B = TTEFFECT6B                                           4900
    PUT FORM MESSAGE = TMSG                                                  4901
    IF IMSG = 23 THEN                                                         4902
        PUT FORM CONTINUE = "Y"                                             4903
    IF IMSG NE 23 THEN                                                         4904
        PUT FORM CONTINUE = TCONTINUE2                                       4905
END RETRIEVE USING                                                            4906
BEGIN                                                                           4907
    TTEFFECT2A = GET FORM EFFECT_2A                                           4908

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

TTEFFECT2B = GET_FORM EFFECT_2B      4909
TTEFFECT3A = GET_FORM EFFECT_3A      4910
TTEFFECT3B = GET_FORM EFFECT_3B      4911
TTEFFECT4A = GET_FORM EFFECT_4A      4912
TTEFFECT4B = GET_FORM EFFECT_4B      4913
TTEFFECT5A = GET_FORM EFFECT_5A      4914
TTEFFECT5B = GET_FORM EFFECT_5B      4915
TTEFFECT6A = GET_FORM EFFECT_6A      4916
TTEFFECT6B = GET_FORM EFFECT_6B      4917
TCONTINUE2 = GET_FORM CONTINUE      4918
END                                  4919

```

```

=====
IF TCONTINUE2 IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH WERE
INADVERTENTLY ENTERED IN TTDESC1, TTDESC2, TTDESC3, TTEFFECT1A,
TTEFFECT1B, TTEFFECT2A, TTEFFECT2B, TTEFFECT3A, TTEFFECT3B,
TTEFFECT4A, TTEFFECT4B, TTEFFECT5A, TTEFFECT5B, TTEFFECT6A AND
TTEFFECT6B ARE REMOVED
=====

```

```

IF TCONTINUE2 NE "A" THEN
  BEGIN
    I = 1
    WHILE I LE 15
      BEGIN
        IF I = 1 THEN TEMP1 = TTDESC1      4938
        IF I = 2 THEN TEMP1 = TTDESC2      4939
        IF I = 3 THEN TEMP1 = TTDESC3      4940
        IF I = 4 THEN TEMP1 = TTEFFECT1A   4941
        IF I = 5 THEN TEMP1 = TTEFFECT1B   4942
        IF I = 6 THEN TEMP1 = TTEFFECT2A   4943
        IF I = 7 THEN TEMP1 = TTEFFECT2B   4944
        IF I = 8 THEN TEMP1 = TTEFFECT3A   4945
        IF I = 9 THEN TEMP1 = TTEFFECT3B   4946
        IF I = 10 THEN TEMP1 = TTEFFECT4A  4947
        IF I = 11 THEN TEMP1 = TTEFFECT4B  4948
        IF I = 12 THEN TEMP1 = TTEFFECT5A  4949
        IF I = 13 THEN TEMP1 = TTEFFECT5B  4950
        IF I = 14 THEN TEMP1 = TTEFFECT6A  4951
        IF I = 15 THEN TEMP1 = TTEFFECT6B  4952
        IF TEMP1 NE " " THEN
          BEGIN
            J = 1
            WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " "
              BEGIN
                J = J + 1
              END
          END
        END
      END
    END
  END

```


Datatrieve Procedure FM_MODIFY_1 (cont.)

```

IF I = 7 THEN TEMP1 = TTEFFECT5A      5011
IF I = 8 THEN TEMP1 = TTEFFECT6A      5012
IF TEMP1 NE " " THEN                  5013
  BEGIN                               5014
    J = 80                            5015
    WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " " 5016
      BEGIN                           5017
        J = J - 1                     5018
      END                             5019
    IF FN$STR_EXTRACT(TEMP1,          5020
      J, 1) = "-" AND                 5021
      FN$STR_EXTRACT(TEMP1,          5022
        J - 1, 1) NE " " AND         5023
      FN$STR_EXTRACT(TEMP1,          5024
        J - 2, 2) NE " -" THEN      5025
      BEGIN                           5026
        FLG4 = "N"                   5027
        IF I = 1 THEN MSG = 13       5028
        IF I = 2 THEN MSG = 14       5029
        IF I = 3 THEN MSG = 15       5030
        IF I = 4 THEN MSG = 16       5031
        IF I = 5 THEN MSG = 17       5032
        IF I = 6 THEN MSG = 18       5033
        IF I = 7 THEN MSG = 19       5034
        IF I = 8 THEN MSG = 20       5035
      END                             5036
    END                               5037
    I = I + 1                         5038
  END                                 5039
IF FLG4 = "Y" THEN                    5040
  BEGIN                               5041
    IF TCONTINUE2 NE "Y" AND          5042
    TCONTINUE2 NE "N" AND            5043
    TCONTINUE2 NE "A" THEN           5044
      BEGIN                           5045
        FLG4 = "N"                   5046
        MSG = 24                     5047
      END                             5048
    END                               5049
  END                                 5050
END                                   5051
!                                     5052
!                                     5053
! =====                           5054
! IF TCONTINUE2 IS NOT EQUAL TO "A", THE VERIFIED DATA FOR THE 5055
! DESCRIPTION AND EFFECTS IS CONCATENATED INTO THE COMPOSITE VARIABLES 5056
! TTDESC AND TTEFFECT1 THROUGH TTEFFECT6 5057
!                                     5058
! =====                           5059
!                                     5060
!                                     5061

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

IF TCONTINUE2 NE "A" THEN                                5062
  BEGIN                                                    5063
    TDESC = TDESC1                                         5064
    IF TDESC = " " THEN                                     5065
      TDESC = TDESC2 ELSE                                   5066
      TDESC = TDESC || " " | TDESC2                       5067
    IF TDESC = " " THEN                                     5068
      TDESC = TDESC3 ELSE                                   5069
      TDESC = TDESC || " " | TDESC3                       5070
    TTEFFECT1 = TTEFFECT1A                                  5071
    IF TTEFFECT1 = " " THEN                                  5072
      TTEFFECT1 = TTEFFECT1B ELSE                           5073
      TTEFFECT1 = TTEFFECT1 || " " | TTEFFECT1B            5074
    TTEFFECT2 = TTEFFECT2A                                  5075
    IF TTEFFECT2 = " " THEN                                  5076
      TTEFFECT2 = TTEFFECT2B ELSE                           5077
      TTEFFECT2 = TTEFFECT2 || " " | TTEFFECT2B            5078
    TTEFFECT3 = TTEFFECT3A                                  5079
    IF TTEFFECT3 = " " THEN                                  5080
      TTEFFECT3 = TTEFFECT3B ELSE                           5081
      TTEFFECT3 = TTEFFECT3 || " " | TTEFFECT3B            5082
    TTEFFECT4 = TTEFFECT4A                                  5083
    IF TTEFFECT4 = " " THEN                                  5084
      TTEFFECT4 = TTEFFECT4B ELSE                           5085
      TTEFFECT4 = TTEFFECT4 || " " | TTEFFECT4B            5086
    TTEFFECT5 = TTEFFECT5A                                  5087
    IF TTEFFECT5 = " " THEN                                  5088
      TTEFFECT5 = TTEFFECT5B ELSE                           5089
      TTEFFECT5 = TTEFFECT5 || " " | TTEFFECT5B            5090
    TTEFFECT6 = TTEFFECT6A                                  5091
    IF TTEFFECT6 = " " THEN                                  5092
      TTEFFECT6 = TTEFFECT6B ELSE                           5093
      TTEFFECT6 = TTEFFECT6 || " " | TTEFFECT6B            5094
  END                                                       5095
!                                                           5096
!                                                           5097
!=====                                                    5098
! IF TCONTINUE2 IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE  5099
! VARIABLES TTEFFECT1, TTEFFECT2, TTEFFECT3, TTEFFECT4, TTEFFECT5 AND  5100
! TTEFFECT6 SO THAT THE FIRST VALUE WHICH IS NOT BLANK IS TTEFFECT1, THE  5101
! SECOND VALUE WHICH IS NOT BLANK IS TTEFFECT2, ETC.              5102
!=====                                                    5103
!                                                           5104
!=====                                                    5105
!                                                           5106
IF TCONTINUE2 NE "A" THEN                                5107
  BEGIN                                                    5108
    TEMP6 = TTEFFECT1 || "$#" | TTEFFECT2 || "$#" | TTEFFECT3 || "$#" |  5109
            TTEFFECT4 || "$#" | TTEFFECT5 || "$#" | TTEFFECT6 || "$#" |  5110
    I = 1                                                    5111
    WHILE I LE 6                                             5112

```


Datatrieve Procedure FM_MODIFY_1 (cont.)

```

BEGIN                                                    5113
  J = 1                                                    5114
  WHILE J = 1                                              5115
    BEGIN                                                  5116
      J = FN$STR_LOC(TEMP6, "$$")                        5117
      IF J = 0 THEN TTEFFECT = " "                      5118
      IF J = 1 THEN TEMP6 = FN$STR_EXTRACT(TEMP6, 4, 981) 5119
      IF J GT 1 THEN                                      5120
        BEGIN                                              5121
          TTEFFECT = FN$STR_EXTRACT(TEMP6, 1, J - 1)      5122
          TEMP6 = FN$STR_EXTRACT(TEMP6, J, 984 - J + 1)  5123
        END                                                5124
      END                                                  5125
      IF I = 1 THEN TTEFFECT1 = TTEFFECT                 5126
      IF I = 2 THEN TTEFFECT2 = TTEFFECT                 5127
      IF I = 3 THEN TTEFFECT3 = TTEFFECT                 5128
      IF I = 4 THEN TTEFFECT4 = TTEFFECT                 5129
      IF I = 5 THEN TTEFFECT5 = TTEFFECT                 5130
      IF I = 6 THEN TTEFFECT6 = TTEFFECT                 5131
      I = I + 1                                           5132
    END                                                    5133
  END                                                      5134
END                                                        5135
!                                                         5136
! ===== 5137
! IF TCONTINUE2 IS NOT EQUAL TO "A" AND NEW DATA HAS BEEN ENTERED IN AT 5138
! LEAST ONE OF THE RECORD FIELDS, THIS SECTION PRINTS THE INITIAL RECORD 5139
! DATA TO THE LOG FILE, MODIFIES THE RECORD AND PRINTS THE MODIFIED      5140
! RECORD DATA TO THE LOG FILE (HIGHLIGHTING THE CHANGED FIELDS)          5141
! ===== 5142
!                                                         5143
! IF TCONTINUE2 NE "A" AND 5144
! (DESCRIPTION NE TTDESC      OR 5145
! EFFECT1      NE TTEFFECT1  OR 5146
! EFFECT2      NE TTEFFECT2  OR 5147
! EFFECT3      NE TTEFFECT3  OR 5148
! EFFECT4      NE TTEFFECT4  OR 5149
! EFFECT5      NE TTEFFECT5  OR 5150
! EFFECT6      NE TTEFFECT6) THEN 5151
! BEGIN 5152
!                                                         5153
! ===== 5154
! THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG 5155
! FILE 5156
! ===== 5157
!                                                         5158
! IF TCONTINUE2 NE "A" AND 5159
! (DESCRIPTION NE TTDESC      OR 5160
! EFFECT1      NE TTEFFECT1  OR 5161
! EFFECT2      NE TTEFFECT2  OR 5162
! EFFECT3      NE TTEFFECT3  OR 5163
! EFFECT4      NE TTEFFECT4  OR
! EFFECT5      NE TTEFFECT5  OR
! EFFECT6      NE TTEFFECT6) THEN
! BEGIN
! =====
! THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG
! FILE
! =====

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

KCNT = KCNT + 1
PRINT NEW_PAGE, COL 1, "RECORD NO. ", SPACE 0,
      KCNT (-) USING 9(4), SKIP 2,
      "===== ", SPACE 0,
      "===== ", SKIP 1,
      "===== ", SPACE 0,
      "===== ", SKIP 2,
COL 6, "DATE_CREATED      :", SPACE 1,
      DATE_CREATED (-) USING X(23), SKIP 1,
COL 6, "FMCODE           :", SPACE 1,
      SOURCE_SYSTEM      (-) USING X(4), SPACE 1,
      SOURCE_MODULE      (-) USING 9(4), SPACE 1,
      FAILURE_MODE       (-) USING X(2), SPACE 1,
      FAILURE_SUBMODE    (-) USING X(2), SPACE 1,
      ACCOMPLICE_SYSTEM  (-) USING X(4), SPACE 1,
      ACCOMPLICE_MODULE  (-) USING 9(4), SKIP 1,
COL 6, "DESCRIPTION      :", SPACE 1,
      DESCRIPTION (-) USING T(52), SKIP 1,
COL 6, "EFFECT1          :", SPACE 1,
      EFFECT1 (-) USING T(52), SKIP 1,
COL 6, "EFFECT2          :", SPACE 1,
      EFFECT2 (-) USING T(52), SKIP 1,
COL 6, "EFFECT3          :", SPACE 1,
      EFFECT3 (-) USING T(52), SKIP 1,
COL 6, "EFFECT4          :", SPACE 1,
      EFFECT4 (-) USING T(52), SKIP 1,
COL 6, "EFFECT5          :", SPACE 1,
      EFFECT5 (-) USING T(52), SKIP 1,
COL 6, "EFFECT6          :", SPACE 1,
      EFFECT6 (-) USING T(52), SKIP 1,
COL 6, "DATE_LAST_MODIFIED :", SPACE 1,
      DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,
COL 6, "MODIFYING_PROCEDURE :", SPACE 1,
      MODIFYING_PROCEDURE (-) USING X(20)

```

```

=====
THIS SECTION STORES THE INITIAL RECORD VALUES FOR USE IN
HIGHLIGHTING THE FIELDS WHICH HAVE BEEN MODIFIED
=====

```

```

HDCREATED = DATE_CREATED
HFMCODE   = FMCODE
HDESC     = DESCRIPTION
HEFFECT1  = EFFECT1
HEFFECT2  = EFFECT2
HEFFECT3  = EFFECT3
HEFFECT4  = EFFECT4
HEFFECT5  = EFFECT5

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

HEFFECT6 = EFFECT6                                5215
HDLASTMOD = DATE_LAST_MODIFIED                    5216
HMODPROC = MODIFYING_PROCEDURE                    5217
                                                    5218
                                                    5219
===== 5220
THIS SECTION MODIFIES THE FAILURE MODE RECORD USING THE VERIFIED 5221
DATA ENTERED ON THE MODIFY FAILURE MODES FORM                    5222
===== 5223
                                                    5224
                                                    5225
                                                    5226
CAL = "NOW"                                         5227
MODIFY USING                                       5228
BEGIN                                             5229
    DESCRIPTION = TTDESC                           5230
    EFFECT1 = TTEFFECT1                           5231
    EFFECT2 = TTEFFECT2                           5232
    EFFECT3 = TTEFFECT3                           5233
    EFFECT4 = TTEFFECT4                           5234
    EFFECT5 = TTEFFECT5                           5235
    EFFECT6 = TTEFFECT6                           5236
    DATE_LAST_MODIFIED = CAL                       5237
    MODIFYING_PROCEDURE = "FM_MODIFY"             5238
END                                                5239
                                                    5240
                                                    5241
===== 5242
THIS SECTION PRINTS THE MODIFIED RECORD DATA TO THE SESSION LOG 5243
FILE AND HIGHLIGHTS THE FIELDS WHICH CONTAIN NEW INFORMATION    5244
===== 5245
                                                    5246
                                                    5247
                                                    5248
PRINT SKIP 1,                                     5249
"=====", SPACE 0,                                  5250
"=====", SKIP 2,                                   5251
COL 1, CHOICE                                       5252
    DATE_CREATED = HDCREATED THEN " "              5253
    ELSE "****"                                    5254
END CHOICE,                                         5255
SPACE 2, "DATE_CREATED :", SPACE 1,               5256
    DATE_CREATED (-) USING X(23), SKIP 1,          5257
COL 1, CHOICE                                       5258
    FMCODE = HFMCODE THEN " "                     5259
    ELSE "****"                                    5260
END CHOICE,                                         5261
SPACE 2, "FMCODE :", SPACE 1,                     5262
    SOURCE_SYSTEM (-) USING X(4), SPACE 1,         5263
    SOURCE_MODULE (-) USING 9(4), SPACE 1,         5264
    FAILURE_MODE (-) USING X(2), SPACE 1,          5265

```

Datatrieve Procedure FM_MODIFY_1 (cont.)

	FAILURE SUBMODE (-) USING X(2), SPACE 1,	5266
	ACCOMPLICE_SYSTEM (-) USING X(4), SPACE 1,	5267
	ACCOMPLICE_MODULE (-) USING 9(4), SKIP 1,	5268
COL 1, CHOICE		5269
	DESCRIPTION = HDESC THEN " "	5270
	ELSE "****"	5271
	END CHOICE,	5272
	SPACE 2, "DESCRIPTION :", SPACE 1,	5273
	DESCRIPTION (-) USING T(52), SKIP 1,	5274
COL 1, CHOICE		5275
	EFFECT1 = HEFFECT1 THEN " "	5276
	ELSE "****"	5277
	END CHOICE,	5278
	SPACE 2, "EFFECT1 :", SPACE 1,	5279
	EFFECT1 (-) USING T(52), SKIP 1,	5280
COL 1, CHOICE		5281
	EFFECT2 = HEFFECT2 THEN " "	5282
	ELSE "****"	5283
	END CHOICE,	5284
	SPACE 2, "EFFECT2 :", SPACE 1,	5285
	EFFECT2 (-) USING T(52), SKIP 1,	5286
COL 1, CHOICE		5287
	EFFECT3 = HEFFECT3 THEN " "	5288
	ELSE "****"	5289
	END CHOICE,	5290
	SPACE 2, "EFFECT3 :", SPACE 1,	5291
	EFFECT3 (-) USING T(52), SKIP 1,	5292
COL 1, CHOICE		5293
	EFFECT4 = HEFFECT4 THEN " "	5294
	ELSE "****"	5295
	END CHOICE,	5296
	SPACE 2, "EFFECT4 :", SPACE 1,	5297
	EFFECT4 (-) USING T(52), SKIP 1,	5298
COL 1, CHOICE		5299
	EFFECT5 = HEFFECT5 THEN " "	5300
	ELSE "****"	5301
	END CHOICE,	5302
	SPACE 2, "EFFECT5 :", SPACE 1,	5303
	EFFECT5 (-) USING T(52), SKIP 1,	5304
COL 1, CHOICE		5305
	EFFECT6 = HEFFECT6 THEN " "	5306
	ELSE "****"	5307
	END CHOICE,	5308
	SPACE 2, "EFFECT6 :", SPACE 1,	5309
	EFFECT6 (-) USING T(52), SKIP 1,	5310
COL 1, CHOICE		5311
	DATE_LAST_MODIFIED = HDLASTMOD THEN " "	5312
	ELSE "****"	5313
	END CHOICE,	5314
	SPACE 2, "DATE_LAST_MODIFIED :", SPACE 1,	5315
	DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,	5316

Datatrieve Procedure FM_MODIFY_1 (cont.)

```

COL 1, CHOICE
    MODIFYING_PROCEDURE = HMODPROC THEN "    "
    ELSE "***"
END CHOICE,
SPACE 2, "MODIFYING_PROCEDURE :", SPACE 1,
MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,
"===== ", SPACE 0,
"===== ", SKIP 1,
"===== ", SPACE 0,
"===== "
: BELL
END
!
! =====
! IF TCONTINUE2 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN MODIFIED. THIS
! MESSAGE WILL APPEAR IN THE SESSION LOG FILE.
! =====
!
IF TCONTINUE2 = "A" THEN
    BEGIN
        PRINT NEW_PAGE, SKIP 3,
        COL 1, "===== ", SPACE 0,
        "===== ", SKIP 10,
        COL 9, "*****", SKIP 2,
        COL 9, "***** RECORD NOT MODIFIED *****", SKIP 2,
        COL 9, "*****", SKIP 10,
        COL 1, "===== ", SPACE 0,
        "===== "
    END
END-PROCEDURE

```

Datatrieve Procedure FM_STORE

```

DEFINE PROCEDURE FM_STORE                                     5351
!                                                             5352
!                                                             5353
!=====                                                    5354
!                                                             5355
!  VARIABLES ASSOCIATED WITH FIRST FIVE INPUT FIELDS FOR    5356
!  DOMAIN FAILUREMODES:                                     5357
!    1. SOURCE_SYSTEM                                       5358
!    2. SOURCE_MODULE                                       5359
!    3. FAILURE_MODE_SUBMODE                               5360
!    4. ACCOMPLICE_SYSTEM                                   5361
!    5. ACCOMPLICE_MODULE                                   5362
!=====                                                    5363
!                                                             5364
!                                                             5365
DECLARE TSSYSTEM      PIC X(4).                               5366
DECLARE TSMODULE      PIC 9(4).                               5367
DECLARE TTSMODULE     PIC X(4).                               5368
DECLARE TFMSUBM       PIC X(4).                               5369
DECLARE TASYSTEM      PIC X(4).                               5370
DECLARE TAMODULE      PIC 9(4).                               5371
DECLARE TTAMODULE     PIC X(4).                               5372
!                                                             5373
!=====                                                    5374
!                                                             5375
!  VARIABLES ASSOCIATED WITH DESCRIPTION                     5376
!=====                                                    5377
!                                                             5378
!=====                                                    5379
!                                                             5380
DECLARE TDESC1        PIC X(80).                             5381
DECLARE TDESC2        PIC X(80).                             5382
DECLARE TDESC3        PIC X(80).                             5383
DECLARE TDESC         PIC X(242).                            5384
!                                                             5385
!=====                                                    5386
!                                                             5387
!  VARIABLES ASSOCIATED WITH EFFECTS                         5388
!=====                                                    5389
!                                                             5390
!=====                                                    5391
!                                                             5392
DECLARE TEFFECT1A     PIC X(80).                              5393
DECLARE TEFFECT1B     PIC X(80).                              5394
DECLARE TEFFECT2A     PIC X(80).                              5395
DECLARE TEFFECT2B     PIC X(80).                              5396
DECLARE TEFFECT3A     PIC X(80).                              5397
DECLARE TEFFECT3B     PIC X(80).                              5398
DECLARE TEFFECT4A     PIC X(80).                              5399
DECLARE TEFFECT4B     PIC X(80).                              5400
DECLARE TEFFECT5A     PIC X(80).                              5401

```

Datatrieve Procedure FM_STORE (cont.)

DECLARE TEFFECT5B	PIC X(80).	5402
DECLARE TEFFECT6A	PIC X(80).	5403
DECLARE TEFFECT6B	PIC X(80).	5404
DECLARE TEFFECT1	PIC X(161).	5405
DECLARE TEFFECT2	PIC X(161).	5406
DECLARE TEFFECT3	PIC X(161).	5407
DECLARE TEFFECT4	PIC X(161).	5408
DECLARE TEFFECT5	PIC X(161).	5409
DECLARE TEFFECT6	PIC X(161).	5410
DECLARE TTEFFECT	PIC X(161).	5411
!		5412
!		5413
=====		5414
! VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR DESCRIPTION AND		5415
! EFFECTS		5416
!		5417
=====		5418
!		5419
!		5420
DECLARE TEMP	PIC X(80).	5421
DECLARE TTEMP	PIC X(984).	5422
!		5423
!		5424
=====		5425
! VARIABLES USED AS FLAGS OR CONDITION INDICATORS		5426
!		5427
=====		5428
!		5429
!		5430
DECLARE TCONTINUE	PIC X(1).	5431
DECLARE IMSG	PIC 9(2).	5432
DECLARE FLG1	PIC X(1).	5433
DECLARE FLG2	PIC X(1).	5434
!		5435
!		5436
=====		5437
! VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM		5438
!		5439
=====		5440
!		5441
!		5442
DECLARE TMSG	PIC X(80).	5443
DECLARE TMSG1	PIC X(80).	5444
DECLARE TMSG2	PIC X(80).	5445
DECLARE TMSG3	PIC X(80).	5446
DECLARE TMSG4	PIC X(80).	5447
DECLARE TMSG5	PIC X(80).	5448
DECLARE TMSG6	PIC X(80).	5449
DECLARE TMSG7	PIC X(80).	5450
DECLARE TMSG8	PIC X(80).	5451
DECLARE TMSG9	PIC X(80).	5452

Datatrieve Procedure FM_STORE (cont.)

```

DECLARE TMSG9A      PIC X(80).                                5453
DECLARE TMSG10      PIC X(80).                                5454
DECLARE TMSG11      PIC X(80).                                5455
DECLARE TMSG12      PIC X(80).                                5456
DECLARE TMSG13      PIC X(80).                                5457
DECLARE TMSG14      PIC X(80).                                5458
DECLARE TMSG15      PIC X(80).                                5459
DECLARE TMSG16      PIC X(80).                                5460
DECLARE TMSG17      PIC X(80).                                5461
DECLARE TMSG18      PIC X(80).                                5462
!                                                              5463
!                                                              5464
!-----!                                                    5465
!  VARIABLES USED AS COUNTERS                                5466
!-----!                                                    5467
!                                                              5468
!-----!                                                    5469
!                                                              5470
DECLARE ICNT        PIC 9(4).                                5471
DECLARE I           PIC 9(4).                                5472
DECLARE J           PIC 9(4).                                5473
!                                                              5474
!-----!                                                    5475
!-----!                                                    5476
!  VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE 5477
!-----!                                                    5478
!                                                              5479
!-----!                                                    5480
!
DECLARE CAL         USAGE DATE                               5481
                     EDIT_STRING X(23).                     5482
!                                                              5483
!-----!                                                    5484
!-----!                                                    5485
!-----!                                                    5486
!  READY THE APPROPRIATE DOMAINS, INITIALIZE THE MESSAGE VARIABLES AND 5487
!  INITIALIZE THE COUNTER (ICNT) USED FOR NUMBERING THE LOG FILE RECORDS 5488
!-----!                                                    5489
!-----!                                                    5490
!-----!                                                    5491
!-----!                                                    5492
SET ABORT                                                    5493
READY FAILUREMODES      SHARED WRITE                          5494
READY SYSTEMS          SHARED READ                            5495
READY MODULES          SHARED READ                            5496
READY FAILUREMODES FORM SHARED READ                          5497
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 5498
TMSG2 = "SOURCE SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS"    5499
TMSG3 = "SOURCE MODULE IS NOT VALID -- NOT IN DOMAIN MODULES FOR SYSTEM" 5500
TMSG4 = "FAILURE MODE & SUBMODE IS NOT VALID -- NOT IN TABLE " | 5501
        "FAILURE MODE SUBMODE TABLE"                          5502
TMSG5 = "ACCOMPLICE SYSTEM & MODULE NOT REQUIRED FOR THIS FAILURE MODE " | 5503

```


Datatrieve Procedure FM_STORE (cont.)

```

      "& SUBMODE" 5504
TMSG6 = "ACCOMPLICE SYSTEM & MODULE REQUIRED FOR THIS FAILURE MODE & SUBMODE" 5505
TMSG7 = "ACCOMPLICE SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS" 5506
TMSG8 = "ACCOMPLICE MODULE IS NOT VALID -- NOT IN DOMAIN MODULES FOR SYSTEM" 5507
TMSG9 = "FAILURE" 5508
TMSG9A = "IS NOT VALID -- ALREADY IN DOMAIN FAILUREMODES" 5509
TMSG10 = "THE FIRST LINE OF THE FAILURE DESCRIPTION SHOULD NOT END WITH " | 5510
        "A HYPENATED WORD" 5511
TMSG11 = "THE SECOND LINE OF THE FAILURE DESCRIPTION SHOULD NOT END WITH " | 5512
        "A HYPENATED WORD" 5513
TMSG12 = "THE FIRST LINE OF EFFECT 1 SHOULD NOT END WITH A HYPENATED WORD" 5514
TMSG13 = "THE FIRST LINE OF EFFECT 2 SHOULD NOT END WITH A HYPENATED WORD" 5515
TMSG14 = "THE FIRST LINE OF EFFECT 3 SHOULD NOT END WITH A HYPENATED WORD" 5516
TMSG15 = "THE FIRST LINE OF EFFECT 4 SHOULD NOT END WITH A HYPENATED WORD" 5517
TMSG16 = "THE FIRST LINE OF EFFECT 5 SHOULD NOT END WITH A HYPENATED WORD" 5518
TMSG17 = "THE FIRST LINE OF EFFECT 6 SHOULD NOT END WITH A HYPENATED WORD" 5519
TMSG18 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A" 5520
ICNT = 0 5521
! 5522
! 5523
===== 5524
! 5525
! PRIMARY LOOP TO STORE FAILUREMODES 5526
! 5527
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS: 5528
! 1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR 5529
!   CORRECTION OF INVALID INFORMATION 5530
! 2. IF TCONTINUE NE "A", SECTION TO CONCATENATE THE INPUT VARIABLES 5531
!   ASSOCIATED WITH DESCRIPTION AND EFFECTS 5532
! 3. IF TCONTINUE NE "A", SECTION TO REARRANGE THE VARIABLES 5533
!   ASSOCIATED WITH EFFECTS 5534
! 4. IF TCONTINUE NE "A", SECTION TO STORE RECORD IN DOMAIN 5535
!   FAILUREMODES AFTER VALIDATION TESTS HAVE BEEN PASSED 5536
! 5. IF TCONTINUE NE "A", SECTION TO PRINT DATA STORED IN DOMAIN 5537
!   FAILUREMODES FOR INCLUSION IN THE SESSION LOG FILE 5538
! 6. IF TCONTINUE = "A", SECTION TO PRINT MESSAGE THAT DATA CURRENTLY 5539
!   ON FORM HAS NOT BEEN STORED 5540
! 7. IF TCONTINUE = "A", SECTION TO REQUEST RESPONSE TO CONTINUE 5541
!   PROCEDURE OR EXIT TO MENU 5542
! 5543
===== 5544
! 5545
TCONTINUE = "Y" 5546
WHILE TCONTINUE = "Y" 5547
BEGIN 5548
! 5549
! 5550
===== 5551
! 5552
! LOOP TO DISPLAY BLANK TDMS FORMS, RETRIEVE THE DATA ENTERED ON THE 5553
! FORMS, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 5554

```

Datatrieve Procedure FM_STORE (cont.)

```

! | _____ | 5555
! |=====| 5556
! | 5557
FLG1 = "N" 5558
IMSG = 1 5559
WHILE FLG1 = "N" 5560
  BEGIN 5561
    IF IMSG = 1 THEN TMSG = TMSG1 5562
    IF IMSG = 2 THEN TMSG = TMSG2 5563
    IF IMSG = 3 THEN TMSG = TMSG3 || " " | TSSYSTEM 5564
    IF IMSG = 4 THEN TMSG = TMSG4 5565
    IF IMSG = 5 THEN TMSG = TMSG5 5566
    IF IMSG = 6 THEN TMSG = TMSG6 5567
    IF IMSG = 7 THEN TMSG = TMSG7 5568
    IF IMSG = 8 THEN TMSG = TMSG8 || " " | TASYSTEM 5569
    IF IMSG = 9 THEN TMSG = TMSG9 || " " | TSSYSTEM | " " | 5570
      TTSMODULE | " " | TFMSUBM | " " | 5571
      TASYSTEM | " " | TTAMODULE | " " | 5572
      TMSG9A 5573
    IF IMSG = 10 THEN TMSG = TMSG10 5574
    IF IMSG = 11 THEN TMSG = TMSG11 5575
    IF IMSG = 12 THEN TMSG = TMSG12 5576
    IF IMSG = 13 THEN TMSG = TMSG13 5577
    IF IMSG = 14 THEN TMSG = TMSG14 5578
    IF IMSG = 15 THEN TMSG = TMSG15 5579
    IF IMSG = 16 THEN TMSG = TMSG16 5580
    IF IMSG = 17 THEN TMSG = TMSG17 5581
    IF IMSG = 18 THEN TMSG = TMSG18 5582
  ! 5583
  ! 5584
  !===== 5585
  ! THIS SECTION DISPLAYS THE STORE FAILURE MODES FORM AND 5586
  ! RETRIEVES THE DATA ENTERED ON THE FORM 5587
  !===== 5588
  ! 5589
  ! 5590
  FOR FIRST 1 FAILUREMODES_FORM 5591
    BEGIN 5592
      DISPLAY FORM FAILUREMODES STO1 FORM IN 5593
      DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING 5594
      BEGIN 5595
        IF IMSG NE 1 THEN 5596
          BEGIN 5597
            PUT FORM SOURCE_SYSTEM = TSSYSTEM 5598
            PUT FORM SOURCE_MODULE = TSMODULE 5600
            PUT FORM MODE_SUBMODE = TFMSUBM 5601
            PUT FORM ACCOM_SYSTEM = TASYSTEM 5602
            PUT FORM ACCOM_MODULE = TAMODULE 5603
            PUT FORM DESCRIPTION_1 = TDESC1 5604
            PUT FORM DESCRIPTION_2 = TDESC2 5605
          END
        END
      END
    END
  END

```

Datatrieve Procedure FM_STORE (cont.)

```

        PUT_FORM DESCRIPTION_3 = TDESC3           5606
        PUT_FORM EFFECT_1A     = TEFFECT1A        5607
        PUT_FORM EFFECT_1B     = TEFFECT1B        5608
    END                                           5609
    IF IMSG = 1 THEN                             5610
    BEGIN                                         5611
        PUT_FORM ACCOM_SYSTEM = "----"          5612
        PUT_FORM ACCOM_MODULE = 0               5613
    END                                           5614
    PUT_FORM MESSAGE = TMSG                      5615
    END RETRIEVE USING                           5616
    BEGIN                                         5617
        TSSYSTEM = GET_FORM SOURCE_SYSTEM        5618
        TSMODULE = GET_FORM SOURCE_MODULE        5619
        TFMSUBM  = GET_FORM MODE_SUBMODE         5620
        TASYSTEM = GET_FORM ACCOM_SYSTEM         5621
        TAMODULE = GET_FORM ACCOM_MODULE         5622
        TDESC1   = GET_FORM DESCRIPTION_1        5623
        TDESC2   = GET_FORM DESCRIPTION_2        5624
        TDESC3   = GET_FORM DESCRIPTION_3        5625
        TEFFECT1A = GET_FORM EFFECT_1A           5626
        TEFFECT1B = GET_FORM EFFECT_1B           5627
    END                                           5628
    DISPLAY FORM FAILUREMODES STO2 FORM IN        5629
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING  5630
    BEGIN                                         5631
        IF IMSG NE 1 THEN                       5632
        BEGIN                                     5633
            PUT_FORM EFFECT_2A = TEFFECT2A        5634
            PUT_FORM EFFECT_2B = TEFFECT2B        5635
            PUT_FORM EFFECT_3A = TEFFECT3A        5636
            PUT_FORM EFFECT_3B = TEFFECT3B        5637
            PUT_FORM EFFECT_4A = TEFFECT4A        5638
            PUT_FORM EFFECT_4B = TEFFECT4B        5639
            PUT_FORM EFFECT_5A = TEFFECT5A        5640
            PUT_FORM EFFECT_5B = TEFFECT5B        5641
            PUT_FORM EFFECT_6A = TEFFECT6A        5642
            PUT_FORM EFFECT_6B = TEFFECT6B        5643
        END                                     5644
        PUT_FORM MESSAGE = TMSG                  5645
        PUT_FORM CONTINUE = TCONTINUE            5646
    END RETRIEVE USING                           5647
    BEGIN                                         5648
        TEFFECT2A = GET_FORM EFFECT_2A           5649
        TEFFECT2B = GET_FORM EFFECT_2B           5650
        TEFFECT3A = GET_FORM EFFECT_3A           5651
        TEFFECT3B = GET_FORM EFFECT_3B           5652
        TEFFECT4A = GET_FORM EFFECT_4A           5653
        TEFFECT4B = GET_FORM EFFECT_4B           5654
        TEFFECT5A = GET_FORM EFFECT_5A           5655
        TEFFECT5B = GET_FORM EFFECT_5B           5656

```

Datatrieve Procedure FM_STORE (cont.)

```

TEFFECT6A = GET_FORM EFFECT_6A      5657
TEFFECT6B = GET_FORM EFFECT_6B      5658
TCONTINUE = GET_FORM CONTINUE      5659
END                                  5660
END                                  5661
TTSMODULE = TSMODULE                5662
IF TSMODULE LT 1000 THEN TTSMODULE = "0" | TTSMODULE 5663
IF TSMODULE LT 100 THEN TTSMODULE = "0" | TTSMODULE 5664
IF TSMODULE LT 10 THEN TTSMODULE = "0" | TTSMODULE 5665
TTAMODULE = TAMODULE                5666
IF TAMODULE LT 1000 THEN TTAMODULE = "0" | TTAMODULE 5667
IF TAMODULE LT 100 THEN TTAMODULE = "0" | TTAMODULE 5668
IF TAMODULE LT 10 THEN TTAMODULE = "0" | TTAMODULE 5669
!                                  5670
!                                  5671
!                                  5672
!                                  5673
!                                  5674
!                                  5675
!                                  5676
!                                  5677
!                                  5678
!                                  5679
!                                  5680
===== 5681
IF TCONTINUE NE "A" THEN 5682
BEGIN 5683
  I = 1 5684
  WHILE I LE 15 5685
  BEGIN 5686
    IF I = 1 THEN TEMP = TDESC1 5687
    IF I = 2 THEN TEMP = TDESC2 5688
    IF I = 3 THEN TEMP = TDESC3 5689
    IF I = 4 THEN TEMP = TEFFECT1A 5690
    IF I = 5 THEN TEMP = TEFFECT1B 5691
    IF I = 6 THEN TEMP = TEFFECT2A 5692
    IF I = 7 THEN TEMP = TEFFECT2B 5693
    IF I = 8 THEN TEMP = TEFFECT3A 5694
    IF I = 9 THEN TEMP = TEFFECT3B 5695
    IF I = 10 THEN TEMP = TEFFECT4A 5696
    IF I = 11 THEN TEMP = TEFFECT4B 5697
    IF I = 12 THEN TEMP = TEFFECT5A 5698
    IF I = 13 THEN TEMP = TEFFECT5B 5699
    IF I = 14 THEN TEMP = TEFFECT6A 5700
    IF I = 15 THEN TEMP = TEFFECT6B 5701
    IF TEMP NE " " THEN 5702
    BEGIN 5703
      J = 1 5704
      WHILE FN$STR_EXTRACT(TEMP, J, 1) = " " 5705
      BEGIN 5706
        J = J + 1 5707

```

Datatrieve Procedure FM_STORE (cont.)

```

                                END
                                TEMP = FN$STR_EXTRACT(TEMP, J, 80 - J + 1)
                                END
                                IF I = 1 THEN TDESC1 = TEMP
                                IF I = 2 THEN TDESC2 = TEMP
                                IF I = 3 THEN TDESC3 = TEMP
                                IF I = 4 THEN TEFFECT1A = TEMP
                                IF I = 5 THEN TEFFECT1B = TEMP
                                IF I = 6 THEN TEFFECT2A = TEMP
                                IF I = 7 THEN TEFFECT2B = TEMP
                                IF I = 8 THEN TEFFECT3A = TEMP
                                IF I = 9 THEN TEFFECT3B = TEMP
                                IF I = 10 THEN TEFFECT4A = TEMP
                                IF I = 11 THEN TEFFECT4B = TEMP
                                IF I = 12 THEN TEFFECT5A = TEMP
                                IF I = 13 THEN TEFFECT5B = TEMP
                                IF I = 14 THEN TEFFECT6A = TEMP
                                IF I = 15 THEN TEFFECT6B = TEMP
                                I = I + 1
                                END
                                END
=====
                                IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
                                VERIFY THE FOLLOWING:
                                1. TSSYSTEM IS IN DOMAIN SYSTEMS
                                2. TSMODULE IS IN DOMAIN MODULES FOR SYSTEM = TSSYSTEM
                                3. TFMSUBM IS IN TABLE FAILURE MODE SUBMODE TABLE
                                4. IF TFMSUBM IS NOT IN TABLE ACCOMPLICE REQUIRED TABLE,
                                THEN TASYSTEM SHOULD BE BLANK OR "----" AND TAMODULE
                                SHOULD BE 0 (ZERO)
                                5. IF TFMSUBM IS IN TABLE ACCOMPLICE REQUIRED TABLE, THEN
                                TASYSTEM AND TAMODULE MUST HAVE NON-TRIVIAL VALUES
                                6. TASYSTEM, IF REQUIRED, IS IN DOMAIN SYSTEMS
                                7. TAMODULE, IF REQUIRED, IS IN DOMAIN MODULES FOR
                                SYSTEM = TASYSTEM
                                8. TSSYSTEM, TSMODULE, TFMSUBM, TASYSTEM AND TAMODULE DO
                                NOT ALREADY EXIST IN DOMAIN FAILUREMODES
                                9. TDESC1 DOES NOT END WITH A HYPHENATED WORD
                                10. TDESC2 DOES NOT END WITH A HYPHENATED WORD
                                11. TEFFECT1A DOES NOT END WITH A HYPHENATED WORD
                                12. TEFFECT2A DOES NOT END WITH A HYPHENATED WORD
                                13. TEFFECT3A DOES NOT END WITH A HYPHENATED WORD
                                14. TEFFECT4A DOES NOT END WITH A HYPHENATED WORD
                                15. TEFFECT5A DOES NOT END WITH A HYPHENATED WORD
                                16. TEFFECT6A DOES NOT END WITH A HYPHENATED WORD
                                17. TCONTINUE IS "Y", "N" OR "A"
=====

```

Datatrieve Procedure FM_STORE (cont.)

```

!
FLG1 = "Y"
IF TCONTINUE NE "A" THEN
  BEGIN
    FLG2 = "N"
    FOR SYSTEMS WITH SYSTEM = TSSYSTEM
      BEGIN
        FLG2 = "Y"
      END
    IF FLG2 = "N" THEN
      BEGIN
        FLG1 = "N"
        IMMSG = 2
      END
    IF FLG1 = "Y" THEN
      BEGIN
        FLG2 = "N"
        FOR MODULES WITH SYSTEM_MODULE = TSSYSTEM | TTSMODULE
          BEGIN
            FLG2 = "Y"
          END
        IF FLG2 = "N" THEN
          BEGIN
            FLG1 = "N"
            IMMSG = 3
          END
        END
      END
    IF FLG1 = "Y" THEN
      BEGIN
        IF TFMSUBM NOT IN FAILURE_MODE_SUBMODE_TABLE THEN
          BEGIN
            FLG1 = "N"
            IMMSG = 4
          END
        END
      END
    IF FLG1 = "Y" THEN
      BEGIN
        IF TFMSUBM NOT IN ACCOMPLICE_REQUIRED_TABLE AND
          (TASYSYEM NE "-----" OR
           TAMODULE NE 0) THEN
          BEGIN
            FLG1 = "N"
            IMMSG = 5
          END
        END
      END
    IF FLG1 = "Y" THEN
      BEGIN
        IF TFMSUBM IN ACCOMPLICE_REQUIRED_TABLE AND
          (TASYSYEM EQ "-----" OR
           TAMODULE EQ 0) THEN
          BEGIN

```

Datatrieve Procedure FM_STORE (cont.)

FLG1 = "N"	5810
IMSG = 6	5811
END	5812
END	5813
IF FLG1 = "Y" AND	5814
TFMSUBM IN ACCOMPLICE_REQUIRED_TABLE THEN	5815
BEGIN	5816
FLG2 = "N"	5817
FOR SYSTEMS WITH SYSTEM = TASYSTEM	5818
BEGIN	5819
FLG2 = "Y"	5820
END	5821
IF FLG2 = "N" THEN	5822
BEGIN	5823
FLG1 = "N"	5824
IMSG = 7	5825
END	5826
END	5827
IF FLG1 = "Y" AND	5828
TFMSUBM IN ACCOMPLICE_REQUIRED_TABLE THEN	5829
BEGIN	5830
FLG2 = "N"	5831
FOR MODULES WITH SYSTEM_MODULE = TASYSTEM TTAMODULE	5832
BEGIN	5833
FLG2 = "Y"	5834
END	5835
IF FLG2 = "N" THEN	5836
BEGIN	5837
FLG1 = "N"	5838
IMSG = 8	5839
END	5840
END	5841
IF FLG1 = "Y" THEN	5842
BEGIN	5843
FOR FAILUREMODES WITH FMCODE = TSSYSTEM TTSMODULE	5844
TFMSUBM	5845
TASYSTEM TTAMODULE	5846
BEGIN	5847
FLG1 = "N"	5848
IMSG = 9	5849
END	5850
END	5851
IF FLG1 = "Y" THEN	5852
BEGIN	5853
I = 1	5854
WHILE FLG1 = "Y" AND I LE 8	5855
BEGIN	5856
IF I = 1 THEN TEMP = TDESC1	5857
IF I = 2 THEN TEMP = TDESC2	5858
IF I = 3 THEN TEMP = TEFFECT1A	5859
IF I = 4 THEN TEMP = TEFFECT2A	5860

Datatrieve Procedure FM_STORE (cont.)

```

      IF I = 5 THEN TEMP = TEFFECT3A      5861
      IF I = 6 THEN TEMP = TEFFECT4A      5862
      IF I = 7 THEN TEMP = TEFFECT5A      5863
      IF I = 8 THEN TEMP = TEFFECT6A      5864
      IF TEMP NE " " THEN                 5865
      BEGIN                               5866
        J = 80                           5867
        WHILE FNSSTR_EXTRACT(TEMP, J, 1) = " " 5868
        BEGIN                             5869
          J = J - 1                       5870
        END                               5871
        IF FNSSTR_EXTRACT(TEMP,           5872
          J, 1) = "-" AND                 5873
          FNSSTR_EXTRACT(TEMP,           5874
          J - 1, 1) NE " " AND           5875
          FNSSTR_EXTRACT(TEMP,           5876
          J - 2, 2) NE " -" THEN         5877
        BEGIN                             5878
          FLG1 = "N"                     5879
          IF I = 1 THEN MSG = 10         5880
          IF I = 2 THEN MSG = 11         5881
          IF I = 3 THEN MSG = 12         5882
          IF I = 4 THEN MSG = 13         5883
          IF I = 5 THEN MSG = 14         5884
          IF I = 6 THEN MSG = 15         5885
          IF I = 7 THEN MSG = 16         5886
          IF I = 8 THEN MSG = 17         5887
        END                               5888
      END                                 5889
      I = I + 1                          5890
    END                                  5891
  END                                   5892
  IF FLG1 = "Y" THEN                     5893
  BEGIN                                  5894
    IF TCONTINUE NE "Y" AND              5895
    TCONTINUE NE "N" AND                 5896
    TCONTINUE NE "A" THEN                5897
    BEGIN                                5898
      FLG1 = "N"                        5899
      MSG = 18                          5900
    END                                  5901
  END                                    5902
END                                      5903
END                                    5904
END                                    5905
=====                               5906
=====                               5907
=====                               5908
=====                               5909
=====                               5910
=====                               5911

```

IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA FOR THE
 DESCRIPTION AND EFFECTS IS CONCATENATED INTO THE COMPOSITE VARIABLES
 TDESC AND TEFFECT1 THROUGH TEFFECT6

Datatrieve Procedure FM_STORE (cont.)

```

! |-----| 5912
! ===== 5913
! 5914
IF TCONTINUE NE "A" THEN 5915
  BEGIN 5916
    TDESC = TDESC1 5917
    IF TDESC = " " THEN 5918
      TDESC = TDESC2 ELSE 5919
      TDESC = TDESC || " " | TDESC2 5920
    IF TDESC = " " THEN 5921
      TDESC = TDESC3 ELSE 5922
      TDESC = TDESC || " " | TDESC3 5923
    TEFFECT1 = TEFFECT1A 5924
    IF TEFFECT1 = " " THEN 5925
      TEFFECT1 = TEFFECT1B ELSE 5926
      TEFFECT1 = TEFFECT1 || " " | TEFFECT1B 5927
    TEFFECT2 = TEFFECT2A 5928
    IF TEFFECT2 = " " THEN 5929
      TEFFECT2 = TEFFECT2B ELSE 5930
      TEFFECT2 = TEFFECT2 || " " | TEFFECT2B 5931
    TEFFECT3 = TEFFECT3A 5932
    IF TEFFECT3 = " " THEN 5933
      TEFFECT3 = TEFFECT3B ELSE 5934
      TEFFECT3 = TEFFECT3 || " " | TEFFECT3B 5935
    TEFFECT4 = TEFFECT4A 5936
    IF TEFFECT4 = " " THEN 5937
      TEFFECT4 = TEFFECT4B ELSE 5938
      TEFFECT4 = TEFFECT4 || " " | TEFFECT4B 5939
    TEFFECT5 = TEFFECT5A 5940
    IF TEFFECT5 = " " THEN 5941
      TEFFECT5 = TEFFECT5B ELSE 5942
      TEFFECT5 = TEFFECT5 || " " | TEFFECT5B 5943
    TEFFECT6 = TEFFECT6A 5944
    IF TEFFECT6 = " " THEN 5945
      TEFFECT6 = TEFFECT6B ELSE 5946
      TEFFECT6 = TEFFECT6 || " " | TEFFECT6B 5947
  END 5948
! 5949
! ===== 5950
! 5951
! IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE 5952
! VARIABLES TEFFECT1, TEFFECT2, TEFFECT3, TEFFECT4, TEFFECT5 AND 5953
! TEFFECT6 SO THAT THE FIRST VALUE WHICH IS NOT BLANK IS TEFFECT1, 5954
! THE SECOND VALUE WHICH IS NOT BLANK IS TEFFECT2, ETC. 5955
! 5956
! ===== 5957
! 5958
! 5959
IF TCONTINUE NE "A" THEN 5960
  BEGIN 5961
    TTEMP = TEFFECT1 || "$#" | TEFFECT2 || "$#" | TEFFECT3 || "$#" | 5962

```

Datatrieve Procedure FM_STORE (cont.)

```

TEFFECT4 || "$#" | TEFFECT5 || "$#" | TEFFECT6 || "$#" 5963
I = 1 5964
WHILE I LE 6 5965
  BEGIN 5966
    J = 1 5967
    WHILE J = 1 5968
      BEGIN 5969
        J = FN$STR_LOC(TTEMP, "$#") 5970
        IF J = 0 THEN TTEFFECT = " " 5971
        IF J = 1 THEN TTEMP = FN$STR_EXTRACT(TTEMP, 4, 981) 5972
        IF J GT 1 THEN 5973
          BEGIN 5974
            TTEFFECT = FN$STR_EXTRACT(TTEMP, 1, J - 1) 5975
            TTEMP = FN$STR_EXTRACT(TTEMP, J, 984 - J + 1) 5976
          END 5977
        END 5978
        IF I = 1 THEN TEFFECT1 = TTEFFECT 5979
        IF I = 2 THEN TEFFECT2 = TTEFFECT 5980
        IF I = 3 THEN TEFFECT3 = TTEFFECT 5981
        IF I = 4 THEN TEFFECT4 = TTEFFECT 5982
        IF I = 5 THEN TEFFECT5 = TTEFFECT 5983
        IF I = 6 THEN TEFFECT6 = TTEFFECT 5984
        I = I + 1 5985
      END 5986
    END 5987
  END 5988
END 5989
===== 5990
IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN 5991
DOMAIN FAILUREMODES 5992
===== 5993
===== 5994
===== 5995
===== 5996
IF TCONTINUE NE "A" THEN 5997
  BEGIN 5998
    CAL = "NOW" 5999
    STORE FAILUREMODES USING 6000
    BEGIN 6001
      DATE_CREATED = CAL 6002
      SOURCE_SYSTEM = TSSYSTEM 6003
      SOURCE_MODULE = TSMODULE 6004
      FAILURE_MODE_SUBMODE = TFMSUBM 6005
      ACCOMPLICE_SYSTEM = TASYSTEM 6006
      ACCOMPLICE_MODULE = TAMODULE 6007
      DESCRIPTION = TDESC 6008
      EFFECT1 = TEFFECT1 6009
      EFFECT2 = TEFFECT2 6010
      EFFECT3 = TEFFECT3 6011
      EFFECT4 = TEFFECT4 6012
      EFFECT5 = TEFFECT5 6013
    END
  END

```

Datatrieve Procedure FM_STORE (cont.)

```

                EFFECT6                = TEFFECT6                6014
            END                6015
        END                6016
!                6017
!                6018
!                6019
!                6020
!                6021
!                6022
!                6023
!                6024
!                6025
!                6026
!                6027
!                6028
!                6029
!                6030
!                6031
!                6032
!                6033
!                6034
!                6035
!                6036
!                6037
!                6038
!                6039
!                6040
!                6041
!                6042
!                6043
!                6044
!                6045
!                6046
!                6047
!                6048
!                6049
!                6050
!                6051
!                6052
!                6053
!                6054
!                6055
!                6056
!                6057
!                6058
!                6059
!                6060
!                6061
!                6062
!                6063
!                6064

=====
IF TCONTINUE IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN
FAILUREMODES IS PRINTED.  THE OUTPUT OF THE PRINT STATEMENTS WILL
BE INCLUDED IN THE SESSION LOG FILE WHICH IS OPENED BY THE CALLING
COMMAND PROCEDURE.
=====

IF TCONTINUE NE "A" THEN
    BEGIN
        ICNT = ICNT + 1
        FOR FAILUREMODES WITH FMCODE = TSSYSTEM | TTSMODULE | TFMSUBM |
            TASYSTEM | TTAMODULE
            PRINT NEW_PAGE, SKIP 3,
                COL 1, "RECORD NO.", SPACE 1,
                    ICNT (-) USING ZZ9, SKIP 1,
                COL 1, "=====", SPACE 0,
                    "=====", SPACE 0,
                    "=====", SPACE 0,
                    "=====", SKIP 2,
                COL 3, "DATE CREATED :", SPACE 1,
                    DATE_CREATED (-) USING X(23), SKIP 1,
                COL 3, "SOURCE SYSTEM MODULE :", SPACE 1,
                    SOURCE_SYSTEM (-) USING X(4), SPACE 1,
                    SOURCE_MODULE (-) USING 9(4), SKIP 1,
                COL 3, "FAILURE MODE SUBMODE :", SPACE 1,
                    FAILURE_MODE_SUBMODE (-) USING X(4), SKIP 1,
                COL 3, "ACCOMPLICE SYSTEM MODULE :", SPACE 1,
                    ACCOMPLICE_SYSTEM (-) USING X(4), SPACE 1,
                    ACCOMPLICE_MODULE (-) USING 9(4), SKIP 1,
                COL 3, "DESCRIPTION :", SPACE 1,
                    DESCRIPTION (-) USING T(50), SKIP 1,
                COL 3, "EFFECT1 :", SPACE 1,
                    EFFECT1 (-) USING T(50), SKIP 1,
                COL 3, "EFFECT2 :", SPACE 1,
                    EFFECT2 (-) USING T(50), SKIP 1,
                COL 3, "EFFECT3 :", SPACE 1,
                    EFFECT3 (-) USING T(50), SKIP 1,
                COL 3, "EFFECT4 :", SPACE 1,
                    EFFECT4 (-) USING T(50), SKIP 1,
                COL 3, "EFFECT5 :", SPACE 1,
                    EFFECT5 (-) USING T(50), SKIP 1,
                COL 3, "EFFECT6 :", SPACE 1,
                    EFFECT6 (-) USING T(50), SKIP 2,
                COL 1, "=====", SPACE 0,

```

Datatrieve Procedure FM_STORE (cont.)

```

"=====", SPACE 0, 6065
"=====", SPACE 0, 6066
"===== 6067

:BELL 6068
END 6069
6070
6071
===== 6072
IF TCONTINUE IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT 6073
THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED. THIS 6074
MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE. 6075
===== 6076
6077
6078
IF TCONTINUE = "A" THEN 6079
BEGIN 6080
PRINT NEW_PAGE, SKIP 3, 6081
COL 1, "=====", SPACE 0, 6082
"===== ", SKIP 4, 6083
COL 9, "*****", SKIP 2, 6084
COL 9, "***** RECORD NOT STORED *****", SKIP 2, 6085
COL 9, "*****", SKIP 4, 6086
COL 1, "===== ", SPACE 0, 6087
"===== 6088
END 6089
6090
6091
6092
===== 6093
IF TCONTINUE EQUALS "A", A RESPONSE IS REQUESTED TO EITHER CONTINUE 6094
THE PROCEDURE TO STORE FAILUREMODES OR EXIT TO THE MENU 6095
===== 6096
6097
6098
6099
IF TCONTINUE = "A" THEN 6100
BEGIN 6101
PRINT NEW_PAGE 6102
:CLRSCRN 6103
TCONTINUE = "X" 6104
WHILE TCONTINUE NE "Y" AND 6105
TCONTINUE NE "N" 6106
BEGIN 6107
PRINT SKIP 2, 6108
"Do you wish to continue entering FAILUREMODES?", 6109
SKIP 1 6110
TCONTINUE = FN$UPCASE(*."Y or N") 6111
PRINT " " 6112
END 6113
END 6114
END 6115

```

Datatrieve Procedure FM_STORE (cont.)

END-PROCEDURE

6116

Datatrieve Procedure HDR

DEFINE PROCEDURE HDR	6117
DECLARE CAL USAGE DATE EDIT_STRING X(20).	6118
CAL = "NOW"	6119
PRINT "<ESC>" "[" "?" "5" "i"	6120
PRINT SPACE 2, CAL (-), SKIP 3	6121
PRINT "<ESC>" "[" "?" "4" "i"	6122
END-PROCEDURE	6123

Datatrieve Procedure MOD_MODIFY

```

DEFINE PROCEDURE MOD_MODIFY
!
!
=====
!
!   VARIABLES ASSOCIATED WITH FIRST THREE INPUT FIELDS FOR
!   DOMAIN MODULES:
!       1. SYSTEM
!       2. MODULE
!       3. SYSTEM_MODULE_NAME
!
=====
!
DECLARE TSYSTEM      PIC X(4).
DECLARE TMODULE      PIC X(4).
DECLARE NMODULE      PIC 9(4).
DECLARE TMODULE1     PIC X(1).
DECLARE TMODULE2     PIC X(1).
DECLARE TMODULE3     PIC X(1).
DECLARE TMODULE4     PIC X(1).
DECLARE TSMNAME      PIC X(80).
DECLARE TTSMNAME     PIC X(80).
!
!
=====
!
!   VARIABLES ASSOCIATED WITH SYSTEM_MODULE_FUNCTION
!
=====
!
DECLARE TSMFUNC1     PIC X(80).
DECLARE TSMFUNC2     PIC X(80).
DECLARE TSMFUNC3     PIC X(80).
DECLARE TSMFUNC      PIC X(242).
DECLARE TTSMFUNC1    PIC X(80).
DECLARE TTSMFUNC2    PIC X(80).
DECLARE TTSMFUNC3    PIC X(80).
DECLARE TTSMFUNC     PIC X(242).
DECLARE TSMFUNCI     PIC X(80).
!
!
=====
!
!   VARIABLES USED AS FLAGS OR CONDITION INDICATORS
!
=====
!
DECLARE TCONTINUE1   PIC X(1).
DECLARE TCONTINUE2   PIC X(1).
DECLARE MSG          PIC 9(2).
DECLARE FLG1         PIC X(1).

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

DECLARE FLG2          PIC X(1).          6175
DECLARE FLG3          PIC X(1).          6176
DECLARE FLG4          PIC X(1).          6177
!                                                              6178
!                                                              6179
!===== 6180
!  VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM 6181
!===== 6182
!                                                              6183
!===== 6184
!                                                              6185
DECLARE TMSG          PIC X(80).          6186
DECLARE TMSG1         PIC X(80).          6187
DECLARE TMSG2         PIC X(80).          6188
DECLARE TMSG3         PIC X(80).          6189
DECLARE TMSG4         PIC X(80).          6190
DECLARE TMSG5         PIC X(80).          6191
DECLARE TMSG6         PIC X(80).          6192
DECLARE TMSG7         PIC X(80).          6193
DECLARE TMSG8         PIC X(80).          6194
DECLARE TMSG9         PIC X(80).          6195
DECLARE TMSG10        PIC X(80).          6196
DECLARE TMSG11        PIC X(80).          6197
DECLARE TMSG12        PIC X(80).          6198
DECLARE TMSG13        PIC X(80).          6199
!                                                              6200
!===== 6201
!  VARIABLES USED AS COUNTERS 6202
!===== 6203
!                                                              6204
!===== 6205
!                                                              6206
DECLARE ICNT          PIC 9(4).          6207
DECLARE JCNT          PIC 9(4).          6208
DECLARE KCNT          PIC 9(4).          6209
DECLARE I             PIC 9(4).          6210
DECLARE J             PIC 9(4).          6211
!                                                              6212
!===== 6213
!  VARIABLES USED TO TEMPORARILY STORE MODULE DATA FOR COMPARISON OF 6214
!  INITIAL AND MODIFIED VALUES 6215
!===== 6216
!                                                              6217
!===== 6218
!                                                              6219
!===== 6220
DECLARE HDCREATED     USAGE DATE          6221
                        EDIT STRING X(23). 6222
DECLARE HSYSTEM       PIC X(4).          6223
DECLARE HMODULE       PIC 9(4).          6224
DECLARE HSMNAME       PIC X(80).          6225

```


Datatrieve Procedure MOD_MODIFY (cont.)

DECLARE HSMFUNC	PIC X(242).	6226
DECLARE HDLASTMOD	USAGE DATE	6227
	EDIT_STRING X(23).	6228
DECLARE HMODPROC	PIC X(20).	6229
!		6230
!		6231
!=====		6232
!		6233
!	VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE	6234
!		6235
!=====		6236
!		6237
DECLARE CAL	USAGE DATE	6238
	EDIT_STRING X(23).	6239
!		6240
!		6241
!=====		6242
!		6243
!	READY THE DOMAINS SYSTEMS, MODULES AND MODULES_FORM	6244
!		6245
!=====		6246
!		6247
SET ABORT		6248
READY MODULES	SHARED WRITE	6249
READY SYSTEMS	SHARED READ	6250
READY MODULES_FORM	SHARED READ	6251
!		6252
!		6253
!=====		6254
!		6255
!	INITIALIZE THE MESSAGE VARIABLES AND INITIALIZE THE COUNTER (KCNT)	6256
!	USED FOR NUMBERING THE LOG FILE RECORDS	6257
!		6258
!=====		6259
!		6260
TMSG1	= "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"	6261
TMSG2	= "DATA MUST BE ENTERED IN AT LEAST ONE OF THE FIELDS OR CONTINUE "	6262
	"MUST BE N"	6263
TMSG3	= "SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS"	6264
TMSG4	= "MODULE IS NOT VALID -- MUST BE NUMERIC IN RANGE 1 TO 9999 INCLUSIVE"	6265
TMSG5	= "SYSTEM & MODULE IS NOT VALID -- NOT IN DOMAIN MODULES"	6266
TMSG6	= "MODULE IS NOT VALID -- NOT IN DOMAIN MODULES"	6267
TMSG7	= "THE FIRST LINE OF THE MODULE FUNCTION SHOULD NOT END WITH "	6268
	"A HYPENATED WORD"	6269
TMSG8	= "THE SECOND LINE OF THE MODULE FUNCTION SHOULD NOT END WITH "	6270
	"A HYPENATED WORD"	6271
TMSG9	= "CONTINUE IS NOT VALID -- MUST BE Y OR N"	6272
TMSG10	= "NO RECORDS HAVE BEEN FOUND WITH THE DATA INDICATED ABOVE"	6273
TMSG11	= "ENTER MODIFICATIONS IN APPROPRIATE FIELDS AND PRESS RETURN KEY"	6274
TMSG12	= "MODULE NAME NOT VALID -- ALREADY EXISTS IN DOMAIN MODULES "	6275
	"FOR SYSTEM"	6276

Datatrieve Procedure MOD_MODIFY (cont.)

```

TMSG13 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"      6277
KCNT    = 0                                                6278
!                                                         6279
!                                                         6280
!=====                                                 6281
!                                                         6282
! PRIMARY LOOP TO MODIFY MODULES                          6283
!                                                         6284
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:          6285
!   1. LOOP TO REQUEST SEARCH DATA, TEST VALUES, PROMPT FOR CORRECTION 6286
!     OF INVALID INFORMATION AND COUNT THE RECORDS WHICH MATCH THE      6287
!     SPECIFIED INPUT FIELDS                                             6288
!   2. IF TCONTINUE1 NE "N", SECTION TO DISPLAY THE MATCHING RECORDS    6289
!     ONE AT A TIME FOR POSSIBLE MODIFICATION (THIS SECTION IS         6290
!     TERMINATED WHEN TCONTINUE2 = "N")                                  6291
!   3. IF TCONTINUE1 = "N", SECTION TO REQUEST RESPONSE TO CONTINUE     6292
!     PROCEDURE OR EXIT TO MENU                                          6293
!=====                                                 6294
!                                                         6295
TCONTINUE1 = "Y"                                           6296
WHILE TCONTINUE1 = "Y"                                     6297
BEGIN                                                       6298
!                                                         6299
!                                                         6300
!=====                                                 6301
!                                                         6302
! LOOP TO DISPLAY BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE    6303
! FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA  6304
!=====                                                 6305
!                                                         6306
!=====                                                 6307
!                                                         6308
!=====                                                 6309
FLG1 = "N"                                                 6310
IMSG = 1                                                    6311
WHILE FLG1 = "N"                                           6312
BEGIN                                                       6313
    IF IMSG = 1 THEN TMSG = TMSG1                          6314
    IF IMSG = 2 THEN TMSG = TMSG2                          6315
    IF IMSG = 3 THEN TMSG = TMSG3                          6316
    IF IMSG = 4 THEN TMSG = TMSG4                          6317
    IF IMSG = 5 THEN TMSG = TMSG5                          6318
    IF IMSG = 6 THEN TMSG = TMSG6                          6319
    IF IMSG = 7 THEN TMSG = TMSG7                          6320
    IF IMSG = 8 THEN TMSG = TMSG8                          6321
    IF IMSG = 9 THEN TMSG = TMSG9                          6322
    IF IMSG = 10 THEN TMSG = TMSG10                        6323
!                                                         6324
!=====                                                 6325
!                                                         6326
! THIS SECTION DISPLAYS THE FIND MODULES FORM AND RETRIEVES           6327

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

!      | THE DATA ENTERED ON THE FORM | 6328
!      |                               | 6329
!      | =====                     | 6330
!      |                               | 6331
FOR FIRST 1 MODULES_FORM 6332
  BEGIN 6333
    DISPLAY_FORM MODULES_FIN_FORM IN 6334
      DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING 6335
    BEGIN 6336
      IF IMSG NE 1 THEN 6337
        BEGIN 6338
          PUT_FORM SYSTEM = TSYSTEM 6339
          PUT_FORM MODULE = TMODULE 6340
          PUT_FORM MODULE_NAME = TSMNAME 6341
          PUT_FORM FUNCTION_1 = TSMFUNC1 6342
          PUT_FORM FUNCTION_2 = TSMFUNC2 6343
          PUT_FORM FUNCTION_3 = TSMFUNC3 6344
        END 6345
        PUT_FORM CONTINUE = TCONTINUE1 6346
        PUT_FORM MESSAGE = TMSG 6347
      END RETRIEVE USING 6348
      BEGIN 6349
        TSYSTEM = GET_FORM SYSTEM 6350
        TMODULE = GET_FORM MODULE 6351
        TSMNAME = GET_FORM MODULE NAME 6352
        TSMFUNC1 = GET_FORM FUNCTION_1 6353
        TSMFUNC2 = GET_FORM FUNCTION_2 6354
        TSMFUNC3 = GET_FORM FUNCTION_3 6355
        TCONTINUE1 = GET_FORM CONTINUE 6356
      END 6357
    END 6358
    TMODULE1 = FN$STR_EXTRACT(TMODULE, 1, 1) 6359
    TMODULE2 = FN$STR_EXTRACT(TMODULE, 2, 1) 6360
    TMODULE3 = FN$STR_EXTRACT(TMODULE, 3, 1) 6361
    TMODULE4 = FN$STR_EXTRACT(TMODULE, 4, 1) 6362
!      |                               | 6363
!      |                               | 6364
!      | =====                     | 6365
!      |                               | 6366
!      | IF TCONTINUE1 IS NOT EQUAL TO "N", ANY LEADING BLANKS WHICH | 6367
!      | WERE INADVERTENTLY ENTERED IN TSMNAME, TSMFUNC1, TSMFUNC2 AND | 6368
!      | TSMFUNC3 ARE REMOVED | 6369
!      |                               | 6370
!      | =====                     | 6371
!      |                               | 6372
IF TCONTINUE1 NE "N" THEN 6373
  BEGIN 6374
    IF TSMNAME NE " " THEN 6375
      BEGIN 6376
        I = 1 6377
        WHILE FN$STR_EXTRACT(TSMNAME, I, 1) = " " 6378

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

        BEGIN
            I = I + 1
        END
        TSMNAME = FN$STR_EXTRACT(TSMNAME, I, 80 - I + 1)
    END
    IF TSMFUNC1 NE " " THEN
        BEGIN
            I = 1
            WHILE FN$STR_EXTRACT(TSMFUNC1, I, 1) = " "
                BEGIN
                    I = I + 1
                END
            TSMFUNC1 = FN$STR_EXTRACT(TSMFUNC1, I, 80 - I + 1)
        END
    IF TSMFUNC2 NE " " THEN
        BEGIN
            I = 1
            WHILE FN$STR_EXTRACT(TSMFUNC2, I, 1) = " "
                BEGIN
                    I = I + 1
                END
            TSMFUNC2 = FN$STR_EXTRACT(TSMFUNC2, I, 80 - I + 1)
        END
    IF TSMFUNC3 NE " " THEN
        BEGIN
            I = 1
            WHILE FN$STR_EXTRACT(TSMFUNC3, I, 1) = " "
                BEGIN
                    I = I + 1
                END
            TSMFUNC3 = FN$STR_EXTRACT(TSMFUNC3, I, 80 - I + 1)
        END
    END
END

```

=====

IF TCONTINUE1 IS NOT EQUAL TO "N", TESTS ARE PERFORMED TO
 VERIFY ANY DATA ENTERED ON THE FORM. AS APPROPRIATE, THE
 PROCEDURE CHECKS ANY OR ALL OF THE FOLLOWING:

1. DATA HAS BEEN ENTERED IN AT LEAST ONE OF THE FIELDS
2. TSYSTEM IS IN DOMAIN SYSTEMS
3. TMODULE IS NUMERIC IN RANGE 1 TO 9999 INCLUSIVE
4. TSYSTEM | TMODULE IS IN DOMAIN MODULES
5. TSMFUNC1 DOES NOT END WITH A HYPENATED WORD
6. TSMFUNC2 DOES NOT END WITH A HYPENATED WORD
7. TCONTINUE1 IS "Y" OR "N"
8. AT LEAST ONE RECORD EXISTS WITH THE DATA SPECIFIED

=====

Datatrieve Procedure MOD_MODIFY (cont.)

```

FLG1 = "Y" 6430
IF TCONTINUE1 NE "N" THEN 6431
  BEGIN 6432
    IF TSYSTEM = " " AND TMODULE = " " AND 6433
      TSMNAME = " " AND 6434
      TSMFUNC1 = " " AND TSMFUNC2 = " " AND TSMFUNC3 = " " THEN 6435
      BEGIN 6436
        FLG1 = "N" 6437
        MSG = 2 6438
      END 6439
    IF FLG1 = "Y" AND 6440
      TSYSTEM NE " " THEN 6441
      BEGIN 6442
        FLG2 = "N" 6443
        FOR FIRST 1 SYSTEMS WITH SYSTEM = TSYSTEM 6444
          BEGIN 6445
            FLG2 = "Y" 6446
          END 6447
          IF FLG2 = "N" THEN 6448
            BEGIN 6449
              FLG1 = "N" 6450
              MSG = 3 6451
            END 6452
          END 6453
        IF FLG1 = "Y" AND 6454
          TMODULE NE " " THEN 6455
          BEGIN 6456
            FLG2 = "N" 6457
            IF (TMODULE1 = " " OR TMODULE1 IN NUMBER_TABLE) AND 6458
              (TMODULE2 = " " OR TMODULE2 IN NUMBER_TABLE) AND 6459
              (TMODULE3 = " " OR TMODULE3 IN NUMBER_TABLE) AND 6460
              (TMODULE4 = " " OR TMODULE4 IN NUMBER_TABLE) THEN 6461
              BEGIN 6462
                NMODULE = TMODULE 6463
                IF NMODULE GT 0 THEN 6464
                  FLG2 = "Y" 6465
                END 6466
              IF FLG2 = "N" THEN 6467
                BEGIN 6468
                  FLG1 = "N" 6469
                  MSG = 4 6470
                END 6471
              END 6472
            IF FLG1 = "Y" AND 6473
              TSYSTEM NE " " AND 6474
              TMODULE NE " " THEN 6475
              BEGIN 6476
                FLG2 = "N" 6477
                TMODULE = NMODULE 6478
                IF NMODULE LT 1000 THEN TMODULE = "0" | TMODULE 6479
                IF NMODULE LT 100 THEN TMODULE = "0" | TMODULE 6480
              END

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

        IF NMODULE LT 10 THEN TMODULE = "0" | TMODULE      6481
        FOR FIRST 1 MODULES WITH                          6482
            SYSTEM_MODULE = TSYSTEM | TMODULE FLG2 = "Y"    6483
        IF FLG2 = "N" THEN                                  6484
            BEGIN                                           6485
                FLG1 = "N"                                   6486
                IMSG = 5                                     6487
            END                                             6488
        END                                                 6489
    IF FLG1 = "Y" AND                                       6490
        TSYSTEM = " " AND                                  6491
        TMODULE NE " " THEN                                6492
        BEGIN                                              6493
            FLG2 = "N"                                       6494
            FOR FIRST 1 MODULES WITH                        6495
                MODULE = NMODULE FLG2 = "Y"                6496
            IF FLG2 = "N" THEN                              6497
                BEGIN                                       6498
                    FLG1 = "N"                              6499
                    IMSG = 6                                6500
                END                                         6501
            END                                             6502
        END                                                 6503
    IF FLG1 = "Y" THEN                                     6504
        BEGIN                                              6505
            I = 1                                           6506
            WHILE FLG1 = "Y" AND I LE 2                     6507
                BEGIN                                       6508
                    IF I = 1 THEN TSMFUNC1 = TSMFUNC1      6509
                    IF I = 2 THEN TSMFUNC1 = TSMFUNC2      6510
                    IF TSMFUNC1 NE " " THEN                 6511
                        BEGIN                               6512
                            J = 80                         6513
                            WHILE FN$STR_EXTRACT(TSMFUNC1, J, 1) = " " 6514
                                BEGIN                       6515
                                    J = J - 1               6516
                                END                           6517
                            IF FN$STR_EXTRACT(TSMFUNC1,    6518
                                J, 1) = "-" AND             6519
                                FN$STR_EXTRACT(TSMFUNC1,    6520
                                J - 1, 1) NE " " AND        6521
                                FN$STR_EXTRACT(TSMFUNC1,    6522
                                J - 2, 2) NE " -" THEN      6523
                                BEGIN                       6524
                                    FLG1 = "N"              6525
                                    IF I = 1 THEN IMSG = 7   6526
                                    IF I = 2 THEN IMSG = 8   6527
                                END                           6528
                            END                               6529
                        END                                   6530
                    END                                     6531
                END
            I = I + 1
        END
    END

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

IF FLG1 = "Y" THEN                                6532
  BEGIN                                           6533
    IF TCONTINUE1 NE "Y" AND                      6534
      TCONTINUE1 NE "N" THEN                    6535
      BEGIN                                       6536
        FLG1 = "N"                               6537
        IMSG = 9                                6538
      END                                         6539
    END                                           6540
  IF FLG1 = "Y" THEN                                6541
    BEGIN                                           6542
      TSMFUNC = TSMFUNC1                          6543
      IF TSMFUNC = " " THEN                      6544
        TSMFUNC = TSMFUNC2 ELSE                  6545
        TSMFUNC = TSMFUNC || " " | TSMFUNC2      6546
      IF TSMFUNC = " " THEN                      6547
        TSMFUNC = TSMFUNC3 ELSE                  6548
        TSMFUNC = TSMFUNC || " " | TSMFUNC3      6549
      END                                         6550
    IF FLG1 = "Y" THEN                                6551
      BEGIN                                           6552
        FLG2 = "N"                                6553
        IF TSYSTEM NE " " AND TMODULE NE " " THEN 6554
          BEGIN                                       6555
            FLG2 = "Y"                               6556
            JCNT = 0                                6557
            FOR MODULES WITH                       6558
              SYSTEM_MODULE = TSYSTEM | TMODULE    6559
            BEGIN                                       6560
              FLG3 = "Y"                               6561
              IF TSMNAME NE " " AND                6562
                SYSTEM_MODULE_NAME NE TSMNAME THEN 6563
                FLG3 = "N"                           6564
              IF TSMFUNC NE " " AND                 6565
                SYSTEM_MODULE_FUNCTION NE TSMFUNC THEN 6566
                FLG3 = "N"                           6567
              IF FLG3 = "Y" THEN JCNT = JCNT + 1      6568
            END                                         6569
          IF JCNT = 0 THEN                            6570
            BEGIN                                       6571
              FLG1 = "N"                               6572
              IMSG = 10                               6573
            END                                         6574
          END                                           6575
        IF FLG2 = "N" THEN                            6576
          BEGIN                                           6577
            JCNT = 0                                6578
            FOR MODULES                               6579
              BEGIN                                       6580
                FLG3 = "Y"                               6581
                IF TSYSTEM NE " " AND                 6582

```

Datatrieve Procedure MOD_MODIFY (cont.)

```

        SYSTEM NE TSYSTEM THEN                                6583
        FLG3 = "N"                                           6584
        IF TMODULE NE " " AND                                6585
        MODULE NE NMODULE THEN                                6586
        FLG3 = "N"                                           6587
        IF TSMNAME NE " " AND                                6588
        SYSTEM_MODULE_NAME NE TSMNAME THEN                    6589
        FLG3 = "N"                                           6590
        IF TSMFUNC NE " " AND                                6591
        SYSTEM_MODULE_FUNCTION NE TSMFUNC THEN                6592
        FLG3 = "N"                                           6593
        IF FLG3 = "Y" THEN JCNT = JCNT + 1                    6594
        END                                                    6595
        IF JCNT = 0 THEN                                       6596
        BEGIN                                                  6597
            FLG1 = "N"                                         6598
            MSG = 10                                           6599
        END                                                    6600
        END                                                    6601
        END                                                    6602
        END                                                    6603
        END                                                    6604
        END                                                    6605
        END                                                    6606
        =====
        IF TCONTINUE1 IS NOT EQUAL TO "N", THIS SECTION DISPLAYS THE
        INDICATED RECORDS ONE AT A TIME AND ALLOWS CORRECTIONS TO BE MADE
        TO ANY OR ALL OF THE RECORDS. THE FIELDS SYSTEM AND MODULE CANNOT
        BE CHANGED SINCE THESE TWO FIELDS UNIQUELY DEFINE THE MODULE.
        =====
        IF TCONTINUE1 NE "N" THEN                                6607
        BEGIN                                                    6608
            TCONTINUE2 = "Y"                                    6609
            FLG2 = "N"                                          6610
            IF TSYSTEM NE " " AND TMODULE NE " " THEN          6611
            BEGIN                                                6612
                FLG2 = "Y"                                       6613
                ICNT = 0                                         6614
                FOR MODULES WITH                                  6615
                SYSTEM_MODULE = TSYSTEM | TMODULE              6616
                BEGIN                                            6617
                    IF TCONTINUE2 = "Y" THEN                    6618
                    BEGIN                                        6619
                        FLG3 = "Y"                                6620
                        IF TSMNAME NE " " AND                    6621
                        SYSTEM_MODULE_NAME NE TSMNAME THEN      6622
                        FLG3 = "N"                                6623
                        IF TSMFUNC NE " " AND                    6624

```


Datatrieve Procedure MOD_MODIFY (cont.)

```

        SYSTEM_MODULE_FUNCTION NE TSMFUNC THEN          6634
        FLG3 = "N"                                       6635
    IF FLG3 = "Y" THEN                                    6636
        BEGIN                                            6637
            ICNT = ICNT + 1                               6638
            :MOD_MODIFY_1                                6639
        END                                              6640
    END                                                  6641
END                                                      6642
END                                                      6643
IF FLG2 = "N" THEN                                       6644
    BEGIN                                                6645
        ICNT = 0                                         6646
        FOR MODULES SORTED BY SYSTEM_MODULE             6647
        BEGIN                                            6648
            IF TCONTINUE2 = "Y" THEN                    6649
                BEGIN                                    6650
                    FLG3 = "Y"                          6651
                    IF TSYSTEM NE " " AND               6652
                       SYSTEM NE TSYSTEM THEN          6653
                        FLG3 = "N"                      6654
                    IF TMODULE NE " " AND               6655
                       MODULE NE NMODULE THEN           6656
                        FLG3 = "N"                      6657
                    IF TSMNAME NE " " AND              6658
                       SYSTEM_MODULE_NAME NE TSMNAME THEN 6659
                        FLG3 = "N"                      6660
                    IF TSMFUNC NE " " AND              6661
                       SYSTEM_MODULE_FUNCTION NE TSMFUNC THEN 6662
                        FLG3 = "N"                      6663
                    IF FLG3 = "Y" THEN                  6664
                        BEGIN                             6665
                            ICNT = ICNT + 1             6666
                            :MOD_MODIFY_1              6667
                        END                                6668
                    END                                  6669
                END                                      6670
            END                                          6671
        END                                            6672
    END                                              6673
END                                                  6674
=====
IF TCONTINUE1 IS EQUAL TO "N", A RESPONSE IS REQUESTED TO EITHER
CONTINUE THE PROCEDURE TO MODIFY MODULES OR EXIT TO THE MENU
=====
IF TCONTINUE1 = "N" THEN                                6682
    BEGIN                                                6683
        PRINT NEW_PAGE                                  6684

```

Datatrieve Procedure MOD_MODIFY (cont.)

:CLRSCRN	6685
TCONTINUE1 = "X"	6686
WHILE TCONTINUE1 NE "Y" AND	6687
TCONTINUE1 NE "N"	6688
BEGIN	6689
PRINT SKIP 2,	6690
"Do you wish to continue modifying", SKIP 1,	6691
"MODULES?", SKIP 1	6692
TCONTINUE1 = FN\$UPCASE(*."Y or N")	6693
PRINT " "	6694
END	6695
END	6696
END	6697
END-PROCEDURE	6698

Datatrieve Procedure MOD_MODIFY_1

```

DEFINE PROCEDURE MOD_MODIFY_1
!
!
=====
! THE FIELDS OF THE INCOMING MODULE RECORD ARE ASSIGNED TO VARIABLES FOR
! DISPLAY AND MODIFICATION. THE FIELDS SYSTEM AND MODULE ARE NOT
! ASSIGNED TO VARIABLES SINCE THESE FIELDS CANNOT BE MODIFIED.
=====
!
TSMNAME = SYSTEM_MODULE_NAME
TSMFUNC = SYSTEM_MODULE_FUNCTION
!
=====
! THIS SECTION SPLITS THE COMPOSITE VARIABLE TSMFUNC INTO VARIABLES
! WHICH ARE 80 CHARACTERS LONG FOR DISPLAY ON THE TDMS FORM
=====
!
I = 1
WHILE I LE 3
  BEGIN
    IF TSMFUNC NE " " THEN
      BEGIN
        J = 1
        WHILE FN$STR_EXTRACT(TSMFUNC, J, 1) = " "
          BEGIN
            J = J + 1
          END
        TSMFUNC = FN$STR_EXTRACT(TSMFUNC, J, 242 - J + 1)
      END
      FLG4 = "N"
      IF FN$STR_EXTRACT(TSMFUNC, 80, 1) = " " OR
        FN$STR_EXTRACT(TSMFUNC, 81, 1) = " " THEN
        BEGIN
          TSMFUNC1 = FN$STR_EXTRACT(TSMFUNC, 1, 80)
          TSMFUNC = FN$STR_EXTRACT(TSMFUNC, 81, 162)
          FLG4 = "Y"
        END
      IF FLG4 = "N" THEN
        BEGIN
          J = 80
          WHILE FN$STR_EXTRACT(TSMFUNC, J, 1) NE " " AND
            J GT 1
            BEGIN
              J = J - 1
            END
          CHOICE

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

        J = 1 THEN TSMFUNC1 = FN$STR_EXTRACT(TTSMFUNC, 1, 80)          6750
        J GT 1 THEN TSMFUNC1 = FN$STR_EXTRACT(TTSMFUNC, 1, J - 1 + 1) 6751
    END_CHOICE                                                         6752
    CHOICE                                                             6753
        J = 1 THEN TTSMFUNC = FN$STR_EXTRACT(TTSMFUNC, 81, 162)       6754
        J GT 1 THEN TTSMFUNC = FN$STR_EXTRACT(TTSMFUNC, J + 1,       6755
            242 - (J + 1) + 1)                                         6756
    END_CHOICE                                                         6757
END                                                                     6758
IF I = 1 THEN TTSMFUNC1 = TSMFUNC1                                    6759
IF I = 2 THEN TTSMFUNC2 = TSMFUNC1                                    6760
IF I = 3 THEN TTSMFUNC3 = TSMFUNC1                                    6761
I = I + 1                                                             6762
END                                                                     6763
!                                                                       6764
!                                                                       6765
!=====                                                             6766
! LOOP TO DISPLAY A MODULE RECORD USING A TDMS FORM, RETRIEVE DATA FROM 6767
! THE FORM, TEST THE INCOMING INFORMATION AND REQUEST CORRECTION OF      6768
! INVALID DATA                                                           6769
!=====                                                             6770
!                                                                       6771
!=====                                                             6772
! FLG4 = "N"                                                             6773
! MSG = 11                                                             6774
! WHILE FLG4 = "N"                                                       6775
!     BEGIN                                                             6776
!         IF MSG = 7 THEN TMSG = TMSG7                                   6777
!         IF MSG = 8 THEN TMSG = TMSG8                                   6778
!         IF MSG = 11 THEN TMSG = TMSG11                                 6779
!         IF MSG = 12 THEN TMSG = TMSG12 || " " | SYSTEM               6780
!         IF MSG = 13 THEN TMSG = TMSG13                                6781
!=====                                                             6782
! THIS SECTION DISPLAYS THE MODIFY MODULES FORM AND RETRIEVES THE      6783
! DATA ENTERED ON THE FORM                                             6784
!=====                                                             6785
!                                                                       6786
!=====                                                             6787
! THIS SECTION DISPLAYS THE MODIFY MODULES FORM AND RETRIEVES THE      6788
! DATA ENTERED ON THE FORM                                             6789
!=====                                                             6790
!                                                                       6791
! DISPLAY FORM MODULES MOD FORM IN                                       6792
! DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING                          6793
! BEGIN                                                                   6794
!     PUT_FORM RECORD_NUMBER = ICNT                                     6795
!     PUT_FORM TOTAL_RECORDS = JCNT                                     6796
!     PUT_FORM SYSTEM = SYSTEM                                           6797
!     PUT_FORM MODULE = MODULE                                           6798
!     PUT_FORM MODULE_NAME = TTSMNAME                                    6799
!     PUT_FORM FUNCTION_1 = TTSMFUNC1                                    6800

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

PUT_FORM FUNCTION_2      = TTSMFUNC2      6801
PUT_FORM FUNCTION_3      = TTSMFUNC3      6802
IF IMSG = 11 THEN        6803
    PUT_FORM CONTINUE     = "Y"           6804
IF IMSG NE 11 THEN        6805
    PUT_FORM CONTINUE     = TCONTINUE2    6806
PUT_FORM MESSAGE          = TMSG          6807
END RETRIEVE USING        6808
BEGIN                     6809
    TTSMNAME = GET_FORM MODULE_NAME      6810
    TTSMFUNC1 = GET_FORM FUNCTION_1      6811
    TTSMFUNC2 = GET_FORM FUNCTION_2      6812
    TTSMFUNC3 = GET_FORM FUNCTION_3      6813
    TCONTINUE2 = GET_FORM CONTINUE       6814
END                           6815
!                               6816
!                               6817
!                               6818
!                               6819
!                               6820
!                               6821
!                               6822
!                               6823
!                               6824
!                               6825
!                               6826
!                               6827
!                               6828
!                               6829
!                               6830
!                               6831
!                               6832
!                               6833
!                               6834
!                               6835
!                               6836
!                               6837
!                               6838
!                               6839
!                               6840
!                               6841
!                               6842
!                               6843
!                               6844
!                               6845
!                               6846
!                               6847
!                               6848
!                               6849
!                               6850
!                               6851
=====
IF TCONTINUE2 IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH WERE
INADVERTENTLY ENTERED IN TTSMNAME, TTSMFUNC1, TTSMFUNC2 AND
TTSMFUNC3 ARE REMOVED
=====
IF TCONTINUE2 NE "A" THEN
    BEGIN
        IF TTSMNAME NE " " THEN
            BEGIN
                I = 1
                WHILE FN$STR_EXTRACT(TTSMNAME, I, 1) = " "
                    BEGIN
                        I = I + 1
                    END
                TTSMNAME = FN$STR_EXTRACT(TTSMNAME, I, 80 - I + 1)
            END
        IF TTSMFUNC1 NE " " THEN
            BEGIN
                I = 1
                WHILE FN$STR_EXTRACT(TTSMFUNC1, I, 1) = " "
                    BEGIN
                        I = I + 1
                    END
                TTSMFUNC1 = FN$STR_EXTRACT(TTSMFUNC1, I, 80 - I + 1)
            END
        IF TTSMFUNC2 NE " " THEN
            BEGIN
                I = 1
                WHILE FN$STR_EXTRACT(TTSMFUNC2, I, 1) = " "
                    BEGIN
                        I = I + 1
                    END
                TTSMFUNC2 = FN$STR_EXTRACT(TTSMFUNC2, I, 80 - I + 1)
            END
        IF TTSMFUNC3 NE " " THEN
            BEGIN
                I = 1
                WHILE FN$STR_EXTRACT(TTSMFUNC3, I, 1) = " "
                    BEGIN
                        I = I + 1
                    END
                TTSMFUNC3 = FN$STR_EXTRACT(TTSMFUNC3, I, 80 - I + 1)
            END
        IF TCONTINUE2 NE "A" THEN
            BEGIN
                I = 1
                WHILE FN$STR_EXTRACT(TCONTINUE2, I, 1) = " "
                    BEGIN
                        I = I + 1
                    END
                TCONTINUE2 = FN$STR_EXTRACT(TCONTINUE2, I, 80 - I + 1)
            END
    END

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

        END
        TTSMFUNC2 = FN$STR_EXTRACT(TTSMFUNC2, I, 80 - I + 1)
    END
    IF TTSMFUNC3 NE " " THEN
        BEGIN
            I = 1
            WHILE FN$STR_EXTRACT(TTSMFUNC3, I, 1) = " "
                BEGIN
                    I = I + 1
                END
            TTSMFUNC3 = FN$STR_EXTRACT(TTSMFUNC3, I, 80 - I + 1)
        END
    END
END
=====
IF TCONTINUE2 IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
VERIFY THE FOLLOWING:
1. TTSMNAME DOES NOT ALREADY EXIST IN DOMAIN MODULES FOR THE
   CURRENT SYSTEM
2. TTSMFUNC1 DOES NOT END WITH A HYPHENATED WORD
3. TTSMFUNC2 DOES NOT END WITH A HYPHENATED WORD
4. TCONTINUE2 IS "Y", "N" OR "A"
=====
FLG4 = "Y"
IF TCONTINUE2 NE "A" THEN
    BEGIN
        IF TTSMNAME NE SYSTEM_MODULE_NAME THEN
            BEGIN
                HSYSTEM = SYSTEM
                HMODULE = MODULE
                FOR MODULES WITH SYSTEM = HSYSTEM AND
                    MODULE NE HMODULE
                    IF SYSTEM_MODULE_NAME = TTSMNAME THEN
                        BEGIN
                            FLG4 = "N"
                            IMSG = 12
                        END
                    END
                END
            BEGIN
                IF FLG4 = "Y" THEN
                    BEGIN
                        I = 1
                        WHILE FLG4 = "Y" AND I LE 2
                            BEGIN
                                IF I = 1 THEN TSMFUNC1 = TTSMFUNC1
                                IF I = 2 THEN TSMFUNC1 = TTSMFUNC2
                                IF TSMFUNC1 NE " " THEN
                                    BEGIN

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

J = 80
WHILE FN$STR_EXTRACT(TSMFUNC1, J, 1) = " "
  BEGIN
    J = J - 1
  END
  IF FN$STR_EXTRACT(TSMFUNC1,
                    J, 1) = "-" AND
    FN$STR_EXTRACT(TSMFUNC1,
                    J - 1, 1) NE " " AND
    FN$STR_EXTRACT(TSMFUNC1,
                    J - 2, 2) NE " -" THEN
    BEGIN
      FLG4 = "N"
      IF I = 1 THEN IMSG = 7
      IF I = 2 THEN IMSG = 8
    END
  END
  I = I + 1
END
IF FLG4 = "Y" THEN
  BEGIN
    IF TCONTINUE2 NE "Y" AND
      TCONTINUE2 NE "N" AND
      TCONTINUE2 NE "A" THEN
      BEGIN
        FLG4 = "N"
        IMSG = 13
      END
    END
  END
END
!
! =====
! IF TCONTINUE2 IS NOT EQUAL TO "A", THE VERIFIED DATA FOR THE
! SYSTEM_MODULE_FUNCTION IS CONCATENATED INTO THE COMPOSITE VARIABLE
! TTSMFUNC
! =====
!
IF TCONTINUE2 NE "A" THEN
  BEGIN
    TTSMFUNC = TTSMFUNC1
    IF TTSMFUNC = " " THEN
      TTSMFUNC = TTSMFUNC2 ELSE
      TTSMFUNC = TTSMFUNC || " " | TTSMFUNC2
    IF TTSMFUNC = " " THEN
      TTSMFUNC = TTSMFUNC3 ELSE
      TTSMFUNC = TTSMFUNC || " " | TTSMFUNC3
  
```

Datatrive Procedure MOD_MODIFY_1 (cont.)

```

END
=====
IF TCONTINUE2 IS NOT EQUAL TO "A" AND NEW DATA HAS BEEN ENTERED IN AT
LEAST ONE OF THE RECORD FIELDS, THIS SECTION PRINTS THE INITIAL RECORD
DATA TO THE LOG FILE, MODIFIES THE RECORD AND PRINTS THE MODIFIED
RECORD DATA TO THE LOG FILE (HIGHLIGHTING THE CHANGED FIELDS)
=====
IF TCONTINUE2 NE "A" AND
(SYSTEM_MODULE_NAME NE TTSMNAME OR
SYSTEM_MODULE_FUNCTION NE TTSMFUNC) THEN
BEGIN
=====
THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG
FILE
=====
KCNT = KCNT + 1
PRINT NEW_PAGE, SKIP 3,
COL 1, "RECORD NO. ", SPACE 0,
      KCNT (-) USING 9(4), SKIP 1,
      "===== ", SPACE 0,
      "===== ", SKIP 1,
      "===== ", SPACE 0,
      "===== ", SKIP 2,
COL 6, "DATE_CREATED      :", SPACE 1,
      DATE_CREATED (-) USING X(23), SKIP 1,
COL 6, "SYSTEM_MODULE      :", SPACE 1,
      SYSTEM (-) USING X(4), SPACE 1,
      MODULE (-) USING 9(4), SKIP 1,
COL 6, "SYSTEM_MODULE_NAME  :", SPACE 1,
      SYSTEM_MODULE_NAME (-) USING T(49), SKIP 1,
COL 6, "SYSTEM_MODULE_FUNCTION :", SPACE 1,
      SYSTEM_MODULE_FUNCTION (-) USING T(49), SKIP 1,
COL 6, "DATE_LAST_MODIFIED   :", SPACE 1,
      DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,
COL 6, "MODIFYING_PROCEDURE  :", SPACE 1,
      MODIFYING_PROCEDURE (-) USING X(20)
=====
THIS SECTION STORES THE INITIAL RECORD VALUES FOR USE IN

```


Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

!      | HIGHLIGHTING THE FIELDS WHICH HAVE BEEN MODIFIED | 7005
!      |=====| 7006
!      | 7007
!      | 7008
HDCREATED = DATE_CREATED 7009
HSYSTEM   = SYSTEM        7010
HMODULE   = MODULE        7011
HSMNAME   = SYSTEM_MODULE_NAME 7012
HSMFUNC   = SYSTEM_MODULE_FUNCTION 7013
HDLASTMOD = DATE_LAST_MODIFIED 7014
HMODPROC  = MODIFYING_PROCEDURE 7015
!      | 7016
!      | 7017
!      |=====| 7018
!      | THIS SECTION MODIFIES THE MODULE RECORD USING THE VERIFIED DATA | 7019
!      | ENTERED ON THE MODIFY MODULES FORM | 7020
!      |=====| 7021
!      | 7022
!      | 7023
!      | 7024
CAL = "NOW" 7025
MODIFY USING 7026
  BEGIN 7027
    SYSTEM_MODULE_NAME = TTSMNAME 7028
    SYSTEM_MODULE_FUNCTION = TTSMFUNC 7029
    DATE_LAST_MODIFIED = CAL 7030
    MODIFYING_PROCEDURE = "MOD_MODIFY" 7031
  END 7032
!      | 7033
!      | 7034
!      |=====| 7035
!      | THIS SECTION PRINTS THE MODIFIED RECORD DATA TO THE SESSION LOG | 7036
!      | FILE AND HIGHLIGHTS THE FIELDS WHICH CONTAIN NEW INFORMATION | 7037
!      |=====| 7038
!      | 7039
!      | 7040
!      | 7041
PRINT SKIP 1, 7042
  COL 1, "=====", SPACE 0, 7043
    "=====", SKIP 2, 7044
  COL 1, CHOICE 7045
    DATE_CREATED = HDCREATED THEN " " 7046
    ELSE "****" 7047
  END CHOICE, 7048
  SPACE 2, "DATE_CREATED :", SPACE 1, 7049
    DATE_CREATED (-) USING X(23), SKIP 1, 7050
  COL 1, CHOICE 7051
    SYSTEM = HSYSTEM AND 7052
    MODULE = HMODULE THEN " " 7053
    ELSE "****" 7054
  END CHOICE, 7055

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

```

SPACE 2, "SYSTEM_MODULE      :", SPACE 1,      7056
SYSTEM (-) USING X(4),      SPACE 1,      7057
MODULE (-) USING 9(4),      SKIP 1,      7058
COL 1, CHOICE      7059
SYSTEM_MODULE_NAME = HSMNAME THEN " "      7060
ELSE "****"      7061
END CHOICE,      7062
SPACE 2, "SYSTEM_MODULE_NAME      :", SPACE 1,      7063
SYSTEM_MODULE_NAME (-) USING T(49), SKIP 1,      7064
COL 1, CHOICE      7065
SYSTEM_MODULE_FUNCTION = HSMFUNC THEN " "      7066
ELSE "****"      7067
END CHOICE,      7068
SPACE 2, "SYSTEM_MODULE_FUNCTION :", SPACE 1,      7069
SYSTEM_MODULE_FUNCTION (-) USING T(49), SKIP 1,      7070
COL 1, CHOICE      7071
DATE_LAST_MODIFIED = HDLASTMOD THEN " "      7072
ELSE "****"      7073
END CHOICE,      7074
SPACE 2, "DATE_LAST_MODIFIED      :", SPACE 1,      7075
DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,      7076
COL 1, CHOICE      7077
MODIFYING_PROCEDURE = HMODPROC THEN " "      7078
ELSE "****"      7079
END CHOICE,      7080
SPACE 2, "MODIFYING_PROCEDURE      :", SPACE 1,      7081
MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,      7082
COL 1, "===== ", SPACE 0,      7083
"===== ", SKIP 1,      7084
"===== ", SPACE 0,      7085
"===== "      7086
:BELL      7087
END      7088
!      7089
!      7090
!=====      7091
!      7092
! IF TCONTINUE2 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT      7093
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN MODIFIED. THIS      7094
! MESSAGE WILL APPEAR IN THE SESSION LOG FILE.      7095
!      7096
!=====      7097
!      7098
IF TCONTINUE2 = "A" THEN      7099
BEGIN      7100
PRINT NEW_PAGE, SKIP 3,      7101
COL 1, "===== ", SPACE 0,      7102
"===== ", SKIP 6,      7103
COL 9, "*****", SKIP 2,      7104
COL 9, "***** RECORD NOT MODIFIED *****", SKIP 2,      7105
COL 9, "*****", SKIP 6,      7106

```

Datatrieve Procedure MOD_MODIFY_1 (cont.)

COL 1, "=====", SPACE 0,	7107
"=====	7108
END	7109
END-PROCEDURE	7110

Datatrieve Procedure MOD_STORE

```

DEFINE PROCEDURE MOD_STORE
!
!
=====
!
!   VARIABLES ASSOCIATED WITH FIRST THREE INPUT FIELDS FOR
!   DOMAIN MODULES:
!       1. SYSTEM
!       2. MODULE
!       3. SYSTEM_MODULE_NAME
!
=====
!
DECLARE TSYSTEM      PIC X(4).
DECLARE TMODULE      PIC 9(4).
DECLARE TTMODULE     PIC X(4).
DECLARE TSMNAME      PIC X(80).
!
!
=====
!
!   VARIABLES ASSOCIATED WITH SYSTEM_MODULE_FUNCTION
!
=====
!
DECLARE TSMFUNC1     PIC X(80).
DECLARE TSMFUNC2     PIC X(80).
DECLARE TSMFUNC3     PIC X(80).
DECLARE TSMFUNCI     PIC X(80).
DECLARE TSMFUNC      PIC X(242).
!
!
=====
!
!   VARIABLES USED AS FLAGS OR CONDITION INDICATORS
!
=====
!
DECLARE TCONTINUE    PIC X(1).
DECLARE IMSG         PIC 9(2).
DECLARE FLG1         PIC X(1).
DECLARE FLG2         PIC X(1).
!
!
=====
!
!   VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM
!
=====
!
DECLARE TMSG         PIC X(80).

```

7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161

Datatrieve Procedure MOD_STORE (cont.)

DECLARE TMSG1	PIC X(80).	7162
DECLARE TMSG2	PIC X(80).	7163
DECLARE TMSG3	PIC X(80).	7164
DECLARE TMSG4	PIC X(80).	7165
DECLARE TMSG5	PIC X(80).	7166
DECLARE TMSG6	PIC X(80).	7167
DECLARE TMSG7	PIC X(80).	7168
DECLARE TMSG8	PIC X(80).	7169
!		7170
=====		7171
!		7172
! VARIABLES USED AS COUNTERS		7173
!		7174
=====		7175
!		7176
DECLARE ICNT	PIC 9(4).	7177
DECLARE JCNT	PIC 9(4).	7178
DECLARE I	PIC 9(4).	7179
DECLARE J	PIC 9(4).	7180
!		7181
!		7182
=====		7183
!		7184
! VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE		7185
!		7186
=====		7187
!		7188
DECLARE CAL	USAGE DATE	7189
	EDIT_STRING X(23).	7190
!		7191
!		7192
=====		7193
!		7194
! READY THE APPROPRIATE DOMAINS, INITIALIZE THE MESSAGE VARIABLES,		7195
! INITIALIZE THE COUNTER (ICNT) USED FOR NUMBERING THE LOG FILE RECORDS		7196
! AND INITIALIZE THE COUNTER (JCNT) USED TO CONTROL THE NUMBER OF		7197
! RECORDS PRINTED ON EACH PAGE OF THE LOG FILE		7198
!		7199
=====		7200
!		7201
SET ABORT		7202
READY MODULES	SHARED WRITE	7203
READY SYSTEMS	SHARED READ	7204
READY MODULES FORM	SHARED READ	7205
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"		7206
TMSG2 = "SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS"		7207
TMSG3 = "MODULE IS NOT VALID -- MUST BE GREATER THAN 0"		7208
TMSG4 = "MODULE IS NOT VALID -- ALREADY EXISTS IN DOMAIN MODULES FOR SYSTEM"		7209
TMSG5 = "MODULE NAME NOT VALID -- ALREADY EXISTS IN DOMAIN MODULES FOR SYSTEM"		7210
TMSG6 = "THE FIRST LINE OF THE MODULE FUNCTION SHOULD NOT END WITH "		7211
"A HYPENATED WORD"		7212

Datatrieve Procedure MOD_STORE (cont.)

```

TMSG7 = "THE SECOND LINE OF THE MODULE FUNCTION SHOULD NOT END WITH " | 7213
      "A HYPENATED WORD" 7214
TMSG8 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A" 7215
ICNT = 0 7216
JCNT = 0 7217
! 7218
! 7219
===== 7220
! 7221
PRIMARY LOOP TO STORE MODULES 7222
! 7223
THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS: 7224
! 7225
1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR 7226
   CORRECTION OF INVALID INFORMATION 7227
! 7228
2. IF TCONTINUE NE "A", SECTION TO STORE RECORD IN DOMAIN 7229
   MODULES AFTER VALIDATION TESTS HAVE BEEN PASSED 7230
! 7231
3. IF TCONTINUE NE "A", SECTION TO PRINT DATA STORED IN DOMAIN 7232
   MODULES FOR INCLUSION IN THE SESSION LOG FILE 7233
! 7234
4. IF TCONTINUE = "A", SECTION TO PRINT MESSAGE THAT DATA CURRENTLY 7235
   ON FORM HAS NOT BEEN STORED 7236
! 7237
5. IF TCONTINUE = "A", SECTION TO REQUEST RESPONSE TO CONTINUE 7238
   PROCEDURE OR EXIT TO MENU 7239
! 7240
===== 7241
! 7242
TCONTINUE = "Y" 7243
WHILE TCONTINUE = "Y" 7244
BEGIN 7245
! 7246
! 7247
===== 7248
! 7249
LOOP TO DISPLAY A BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE 7250
FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 7251
! 7252
===== 7253
! 7254
FLG1 = "N" 7255
IMSG = 1 7256
WHILE FLG1 = "N" 7257
BEGIN 7258
! 7259
IF IMSG = 1 THEN TMSG = TMSG1 7260
IF IMSG = 2 THEN TMSG = TMSG2 7261
IF IMSG = 3 THEN TMSG = TMSG3 7262
IF IMSG = 4 THEN TMSG = TMSG4 || " " | TSYSTEM 7263
IF IMSG = 5 THEN TMSG = TMSG5 || " " | TSYSTEM
IF IMSG = 6 THEN TMSG = TMSG6
IF IMSG = 7 THEN TMSG = TMSG7
IF IMSG = 8 THEN TMSG = TMSG8
!
!

```

Datatrieve Procedure MOD_STORE (cont.)

```

===== 7264
THIS SECTION DISPLAYS THE STORE MODULES FORM AND RETRIEVES 7265
THE DATA ENTERED ON THE FORM 7266
===== 7267
7268
7269
7270
FOR FIRST 1 MODULES_FORM 7271
BEGIN 7272
    DISPLAY FORM MODULES_STO_FORM IN 7273
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING 7274
    BEGIN 7275
        IF IMSG NE 1 THEN 7276
            BEGIN 7277
                PUT_FORM SYSTEM = TSYSTEM 7278
                PUT_FORM MODULE = TMODULE 7279
                PUT_FORM MODULE_NAME = TSMNAME 7280
                PUT_FORM FUNCTION_1 = TSMFUNC1 7281
                PUT_FORM FUNCTION_2 = TSMFUNC2 7282
                PUT_FORM FUNCTION_3 = TSMFUNC3 7283
            END 7284
            PUT_FORM CONTINUE = TCONTINUE 7285
            PUT_FORM MESSAGE = TMSG 7286
        END RETRIEVE USING 7287
        BEGIN 7288
            TSYSTEM = GET_FORM SYSTEM 7289
            TMODULE = GET_FORM MODULE 7290
            TSMNAME = GET_FORM MODULE_NAME 7291
            TSMFUNC1 = GET_FORM FUNCTION_1 7292
            TSMFUNC2 = GET_FORM FUNCTION_2 7293
            TSMFUNC3 = GET_FORM FUNCTION_3 7294
            TCONTINUE = GET_FORM CONTINUE 7295
        END 7296
    END 7297
END 7298
===== 7299
IF TCONTINUE IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH 7300
WERE INADVERTENTLY ENTERED IN TSMNAME, TSMFUNC1, TSMFUNC2 AND 7301
TSMFUNC3 ARE REMOVED 7302
===== 7303
7304
7305
7306
7307
IF TCONTINUE NE "A" THEN 7308
BEGIN 7309
    I = 1 7310
    WHILE FN$STR_EXTRACT(TSMNAME, I, 1) = " " 7311
        BEGIN 7312
            I = I + 1 7313
        END 7314

```

Datatrieve Procedure MOD_STORE (cont.)

```

TSMNAME = FN$STR_EXTRACT(TSMNAME, I, 80 - I + 1) 7315
IF TSMFUNC1 NE "-" THEN 7316
  BEGIN 7317
    I = 1 7318
    WHILE FN$STR_EXTRACT(TSMFUNC1, I, 1) = " " 7319
      BEGIN 7320
        I = I + 1 7321
      END 7322
    TSMFUNC1 = FN$STR_EXTRACT(TSMFUNC1, I, 80 - I + 1) 7323
  END 7324
IF TSMFUNC2 NE " " THEN 7325
  BEGIN 7326
    I = 1 7327
    WHILE FN$STR_EXTRACT(TSMFUNC2, I, 1) = " " 7328
      BEGIN 7329
        I = I + 1 7330
      END 7331
    TSMFUNC2 = FN$STR_EXTRACT(TSMFUNC2, I, 80 - I + 1) 7332
  END 7333
IF TSMFUNC3 NE " " THEN 7334
  BEGIN 7335
    I = 1 7336
    WHILE FN$STR_EXTRACT(TSMFUNC3, I, 1) = " " 7337
      BEGIN 7338
        I = I + 1 7339
      END 7340
    TSMFUNC3 = FN$STR_EXTRACT(TSMFUNC3, I, 80 - I + 1) 7341
  END 7342
END 7343
7344
7345
===== 7346
IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO 7347
VERIFY THE FOLLOWING: 7348
1. TSYSTEM IS IN DOMAIN SYSTEMS 7349
2. TMODULE IS NOT EQUAL TO 0 (ZERO) 7350
3. TMODULE DOES NOT ALREADY EXIST FOR RECORDS IN DOMAIN 7351
   MODULES WITH SYSTEM = TSYSTEM 7352
4. TSMNAME DOES NOT ALREADY EXIST FOR RECORDS IN DOMAIN 7353
   MODULES WITH SYSTEM = TSYSTEM 7354
5. TSMFUNC1 DOES NOT END WITH A HYPENATED WORD 7355
6. TSMFUNC2 DOES NOT END WITH A HYPENATED WORD 7356
7. TCONTINUE IS "Y", "N" OR "A" 7357
7358
===== 7359
7360
7361
FLG1 = "Y" 7362
IF TCONTINUE NE "A" THEN 7363
  BEGIN 7364
    FLG2 = "N" 7365

```


Datatrieve Procedure MOD_STORE (cont.)

FOR SYSTEMS WITH SYSTEM = TSYSTEM	7366
BEGIN	7367
FLG2 = "Y"	7368
END	7369
IF FLG2 = "N" THEN	7370
BEGIN	7371
FLG1 = "N"	7372
IMSG = 2	7373
END	7374
IF FLG1 = "Y" THEN	7375
BEGIN	7376
IF TMODULE = 0 THEN	7377
BEGIN	7378
FLG1 = "N"	7379
IMSG = 3	7380
END	7381
END	7382
IF FLG1 = "Y" THEN	7383
BEGIN	7384
FOR MODULES WITH SYSTEM = TSYSTEM AND	7385
MODULE = TMODULE	7386
BEGIN	7387
FLG1 = "N"	7388
IMSG = 4	7389
END	7390
END	7391
IF FLG1 = "Y" THEN	7392
BEGIN	7393
FOR MODULES WITH SYSTEM	7394
= TSYSTEM AND	7395
SYSTEM_MODULE_NAME = TSMNAME	7396
BEGIN	7397
FLG1 = "N"	7398
IMSG = 5	7399
END	7400
END	7401
IF FLG1 = "Y" THEN	7402
BEGIN	7403
I = 1	7404
WHILE FLG1 = "Y" AND I LE 2	7405
BEGIN	7406
IF I = 1 THEN TSMFUNC1 = TSMFUNC1	7407
IF I = 2 THEN TSMFUNC1 = TSMFUNC2	7408
IF TSMFUNC1 NE " " THEN	7409
BEGIN	7410
J = 80	7411
WHILE FN\$STR_EXTRACT(TSMFUNC1, J, 1) = " "	7412
BEGIN	7413
J = J - 1	7414
END	7415
IF FN\$STR_EXTRACT(TSMFUNC1,	7416
J, 1) = "-" AND	

Datatrieve Procedure MOD_STORE (cont.)

```

                                FN$STR_EXTRACT(TSMFUNC1,          7417
                                                J - 1, 1) NE " " AND 7418
                                FN$STR_EXTRACT(TSMFUNC1,          7419
                                                J - 2, 2) NE " -" THEN 7420
                                BEGIN                                7421
                                    FLG1 = "N"                      7422
                                    IF I = 1 THEN IMMSG = 6         7423
                                    IF I = 2 THEN IMMSG = 7         7424
                                END                                  7425
                                    END                                7426
                                    I = I + 1                        7427
                                END                                  7428
                            END                                    7429
                            IF FLG1 = "Y" THEN                      7430
                                BEGIN                                7431
                                    IF TCONTINUE NE "Y" AND         7432
                                    TCONTINUE NE "N" AND           7433
                                    TCONTINUE NE "A" THEN           7434
                                        BEGIN                          7435
                                            FLG1 = "N"              7436
                                            IMMSG = 8               7437
                                        END                            7438
                                    END                                7439
                                END                                    7440
                            END                                    7441
                        END                                          7442
                    END                                          7443
                END                                          7444
            END                                          7445
        END                                          7446
    END                                          7447
END                                          7448
===== 7449
IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN 7450
DOMAIN MODULES 7451
===== 7452
IF TCONTINUE NE "A" THEN 7453
    BEGIN 7454
        TSMFUNC = TSMFUNC1 7455
        IF TSMFUNC = " " THEN 7456
            TSMFUNC = TSMFUNC2 ELSE 7457
            TSMFUNC = TSMFUNC || " " | TSMFUNC2 7458
        IF TSMFUNC = " " THEN 7459
            TSMFUNC = TSMFUNC3 ELSE 7460
            TSMFUNC = TSMFUNC || " " | TSMFUNC3 7461
        CAL = "NOW" 7462
        STORE MODULES USING 7463
            BEGIN 7464
                DATE_CREATED = CAL 7465
                SYSTEM = TSYSTEM 7466
                MODULE = TMODULE 7467
                SYSTEM_MODULE_NAME = TSMNAME
                SYSTEM_MODULE_FUNCTION = TSMFUNC
            END
        END
    END
END

```

IF TCONTINUE IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN MODULES IS PRINTED. THE OUTPUT OF THE PRINT STATEMENTS WILL BE INCLUDED IN THE SESSION LOG FILE WHICH IS OPENED BY THE CALLING COMMAND PROCEDURE.

```

ICNT = ICNT + 1
JCNT = JCNT + 1
TTMODULE = TMODULE
IF TMODULE LT 1000 THEN TTMODULE = "0" | TTMODULE
IF TMODULE LT 100 THEN TTMODULE = "0" | TTMODULE
IF TMODULE LT 10 THEN TTMODULE = "0" | TTMODULE
IF JCNT = 1 THEN PRINT NEW PAGE
FOR MODULES WITH SYSTEM_MODULE = TSYSTEM | TTMODULE AND
SYSTEM_MODULE NAME = TSMNAME

```

```

PRINT SKIP 3,
COL 1, "RECORD NO.", SPACE 1,
      ICNT (-) USING ZZ9, SKIP 1,
COL 1, "=====", SPACE 0,
      "=====", SPACE 0,
      "=====", SPACE 0,
      "=====", SKIP 2,
COL 3, "DATE_CREATED :", SPACE 1,
      DATE_CREATED (-) USING X(23), SKIP 1,
COL 3, "SYSTEM :", SPACE 1,
      SYSTEM (-) USING X(4), SKIP 1,
COL 3, "MODULE :", SPACE 1,
      MODULE (-) USING 9(4), SKIP 1,
COL 3, "SYSTEM_MODULE_NAME :", SPACE 1,
      SYSTEM_MODULE_NAME (-) USING T(50), SKIP 1,
COL 3, "SYSTEM_MODULE_FUNCTION :", SPACE 1,
      SYSTEM_MODULE_FUNCTION (-) USING T(50), SKIP 2,
COL 1, "=====", SPACE 0,
      "=====", SPACE 0,
      "=====", SPACE 0,
      "====="

```

```
IF JCNT = 3 THEN JCNT = 0
:BELL
```

END

Datatrieve Procedure MOD_STORE (cont.)

```

!
! IF TCONTINUE IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED. THIS
! MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE.
!
=====
7519
7520
7521
7522
7523
7524
7525
7526 IF TCONTINUE = "A" THEN
7527 BEGIN
7528     JCNT = JCNT + 1
7529     IF JCNT = 1 THEN PRINT NEW_PAGE
7530     PRINT SKIP 3,
7531         COL 1, "===== ", SPACE 0,
7532             "===== ", SKIP 2,
7533         COL 9, "*****", SKIP 2,
7534         COL 9, "***** RECORD NOT STORED *****", SKIP 2,
7535         COL 9, "*****", SKIP 2,
7536         COL 1, "===== ", SPACE 0,
7537             "===== "
7538     IF JCNT = 3 THEN JCNT = 0
7539 END
7540
!
!
!
!
=====
7541
7542
7543 IF TCONTINUE EQUALS "A", A RESPONSE IS REQUESTED TO EITHER CONTINUE
7544 THE PROCEDURE TO STORE MODULES OR EXIT TO THE MENU
7545
7546
7547
7548
7549
7550 IF TCONTINUE = "A" THEN
7551 BEGIN
7552     PRINT NEW_PAGE
7553     :CLRSCRN
7554     TCONTINUE = "X"
7555     WHILE TCONTINUE NE "Y" AND
7556         TCONTINUE NE "N"
7557     BEGIN
7558         PRINT SKIP 2,
7559             "Do you wish to continue entering MODULES?",
7560             SKIP 1
7561         TCONTINUE = FN$UPCASE(*."Y or N")
7562         PRINT " "
7563     END
7564 END
7565 END-PROCEDURE

```

Datatrieve Procedure PRNTOFF

DEFINE PROCEDURE PRNTOFF	7566
PRINT "<ESC>" "[" "?" "4" "i"	7567
END-PROCEDURE	7568

Datatrieve Procedure PRNTON

DEFINE PROCEDURE PRNTON

7569

PRINT "<ESC>" || "[" || "?" || "5" || "i"

7570

END-PROCEDURE

7571

Datatrieve Procedure REF_MODIFY

```

DEFINE PROCEDURE REF_MODIFY                                7572
!                                                         7573
!                                                         7574
!=====                                                 7575
!                                                         7576
!  VARIABLES ASSOCIATED WITH AUTHORS                     7577
!=====                                                 7578
!                                                         7579
!                                                         7580
DECLARE TAUTHOR1      PIC X(25).                          7581
DECLARE TAUTHOR2      PIC X(25).                          7582
DECLARE TAUTHOR3      PIC X(25).                          7583
DECLARE TAUTHOR4      PIC X(25).                          7584
DECLARE TTAUTHOR1     PIC X(25).                          7585
DECLARE TTAUTHOR2     PIC X(25).                          7586
DECLARE TTAUTHOR3     PIC X(25).                          7587
DECLARE TTAUTHOR4     PIC X(25).                          7588
DECLARE TAUTHOR       PIC X(25).                          7589
!                                                         7590
!                                                         7591
!=====                                                 7592
!                                                         7593
!  VARIABLES ASSOCIATED WITH DOCUMENT_TITLE              7594
!=====                                                 7595
!                                                         7596
!                                                         7597
DECLARE TTITLE1       PIC X(80).                          7598
DECLARE TTITLE2       PIC X(80).                          7599
DECLARE TTITLE        PIC X(161).                         7600
DECLARE XTITLE1       PIC X(80).                          7601
DECLARE XTITLE2       PIC X(80).                          7602
DECLARE XTITLE        PIC X(161).                         7603
!                                                         7604
!                                                         7605
!=====                                                 7606
!                                                         7607
!  VARIABLES ASSOCIATED WITH DOCUMENT_DATE               7608
!=====                                                 7609
!                                                         7610
!                                                         7611
DECLARE TDOCDATE      PIC X(9).                           7612
DECLARE TDAY          PIC X(2).                           7613
DECLARE TDAY1         PIC X(1).                           7614
DECLARE TDAY2         PIC X(1).                           7615
DECLARE TMONTH        PIC X(3).                           7616
DECLARE TYEAR         PIC X(4).                           7617
DECLARE TYEAR1        PIC X(1).                           7618
DECLARE TYEAR2        PIC X(1).                           7619
DECLARE TYEAR3        PIC X(1).                           7620
DECLARE TYEAR4        PIC X(1).                           7621
DECLARE NDAY          PIC 9(2).                           7622

```

Datatrieve Procedure REF_MODIFY (cont.)

DECLARE NYEAR	PIC 9(4).	7623
!		7624
!		7625
=====		7626
!		7627
!	VARIABLES ASSOCIATED WITH OTHER INPUT FIELDS FOR	7628
!	DOMAIN REFERENCES:	7629
!	1. DOCUMENT_SOURCE	7630
!	2. DOCUMENT_NUMBER	7631
!	3. CONTRACT_NUMBER	7632
!		7633
=====		7634
!		7635
DECLARE TSOURCE	PIC X(30).	7636
DECLARE TDOCNO	PIC X(30).	7637
DECLARE TCONTNO	PIC X(20).	7638
DECLARE TTDONNO	PIC X(30).	7639
DECLARE TTCONTNO	PIC X(20).	7640
!		7641
!		7642
=====		7643
!		7644
!	VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR AUTHORS,	7645
!	DOCUMENT_TITLE, DOCUMENT_SOURCE, DOCUMENT_NUMBER AND CONTRACT_NUMBER	7646
!		7647
=====		7648
!		7649
DECLARE TEMP1	PIC X(112).	7650
DECLARE TEMP2	PIC X(80).	7651
!		7652
!		7653
=====		7654
!		7655
!	VARIABLES USED AS FLAGS OR CONDITION INDICATORS	7656
!		7657
=====		7658
!		7659
DECLARE TCONTINUE1	PIC X(1).	7660
DECLARE TCONTINUE2	PIC X(1).	7661
DECLARE IMSG	PIC 9(2).	7662
DECLARE FLG1	PIC X(1).	7663
DECLARE FLG2	PIC X(1).	7664
DECLARE FLG3	PIC X(1).	7665
DECLARE FLG4	PIC X(1).	7666
!		7667
!		7668
=====		7669
!		7670
!	VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM	7671
!		7672
=====		7673

Datatrieve Procedure REF_MODIFY (cont.)

```

!
DECLARE TMSG          PIC X(80).
DECLARE TMSG1         PIC X(80).
DECLARE TMSG2         PIC X(80).
DECLARE TMSG3         PIC X(80).
DECLARE TMSG4         PIC X(80).
DECLARE TMSG5         PIC X(80).
DECLARE TMSG6         PIC X(80).
DECLARE TMSG7         PIC X(80).
DECLARE TMSG7A        PIC X(80).
DECLARE TMSG8         PIC X(80).
DECLARE TMSG9         PIC X(80).
DECLARE TMSG10        PIC X(80).
DECLARE TMSG11        PIC X(80).
DECLARE TMSG12        PIC X(80).
!
=====
!
!  VARIABLES USED AS COUNTERS
!
=====
!
DECLARE ICNT          PIC 9(4).
DECLARE JCNT          PIC 9(4).
DECLARE KCNT          PIC 9(4).
DECLARE I             PIC 9(4).
DECLARE J             PIC 9(4).
!
!
=====
!
!  VARIABLES USED TO TEMPORARILY STORE REFERENCE DATA FOR COMPARISON OF
!  INITIAL AND MODIFIED VALUES
!
=====
!
DECLARE HDCREATED      USAGE DATE
                      EDIT STRING X(23).
DECLARE HREFNO         PIC X(5).
DECLARE HAUTHOR1       PIC X(25).
DECLARE HAUTHOR2       PIC X(25).
DECLARE HAUTHOR3       PIC X(25).
DECLARE HAUTHOR4       PIC X(25).
DECLARE HTITLE         PIC X(161).
DECLARE HSOURCE        PIC X(30).
DECLARE HDOCNO         PIC X(30).
DECLARE HDOCDATE       PIC X(11).
DECLARE HCONTNO        PIC X(20).
DECLARE HDLASTMOD      USAGE DATE
                      EDIT STRING X(23).
DECLARE HMODPROC       PIC X(20).

```

Datatrieve Procedure REF_MODIFY (cont.)

```

!
!
=====
!  VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE
!
=====
!
DECLARE CAL      USAGE DATE
                EDIT_STRING X(23).
!
!
=====
!  READY THE DOMAINS REFERENCES AND REFERENCES_FORM
!
=====
!
SET ABORT
READY REFERENCES      SHARED WRITE
READY REFERENCES_FORM SHARED READ
!
!
=====
!  INITIALIZE THE MESSAGE VARIABLES AND INITIALIZE THE COUNTER (KCNT)
!  USED FOR NUMBERING THE LOG FILE RECORDS
!
=====
!
TMSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"
TMSG2 = "DATA MUST BE ENTERED IN AT LEAST ONE OF THE FIELDS OR CONTINUE " |
        "MUST BE N"
TMSG3 = "THE FIRST LINE OF THE DOCUMENT TITLE SHOULD NOT END WITH " |
        "A HYPHENATED WORD"
TMSG4 = "ORIGINATING ORGANIZATION IS NOT VALID -- NOT IN TABLE " |
        "REFERENCE_SOURCE_TABLE"
TMSG5 = "DOCUMENT DATE IS NOT VALID -- DAY MUST BE NUMERIC IN RANGE 0 TO " |
        "31 INCLUSIVE"
TMSG6 = "DOCUMENT DATE IS NOT VALID -- MONTH MUST BE JAN, FEB, MAR, ETC."
TMSG7 = "DOCUMENT DATE IS NOT VALID --"
TMSG7A = "EXCEEDS THE NUMBER OF DAYS IN"
TMSG8 = "DOCUMENT DATE NOT VALID -- YEAR MUST BE NUMERIC IN RANGE 1960 " |
        "TO 9999 INCLUSIVE"
TMSG9 = "CONTINUE IS NOT VALID -- MUST BE Y OR N"
TMSG10 = "NO RECORDS HAVE BEEN FOUND WITH THE DATA INDICATED ABOVE"
TMSG11 = "ENTER MODIFICATIONS IN APPROPRIATE FIELDS AND PRESS RETURN KEY"
TMSG12 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"
KCNT = 0
!
!

```

7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775

Datatrieve Procedure REF_MODIFY (cont.)

```

===== 7776
! PRIMARY LOOP TO MODIFY REFERENCES 7777
! 7778
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS: 7779
! 1. LOOP TO REQUEST SEARCH DATA, TEST VALUES, PROMPT FOR CORRECTION 7780
!   OF INVALID INFORMATION AND COUNT THE RECORDS WHICH MATCH THE 7781
!   SPECIFIED INPUT FIELDS 7782
! 2. IF TCONTINUE1 NE "N", SECTION TO DISPLAY THE MATCHING RECORDS 7783
!   ONE AT A TIME FOR POSSIBLE MODIFICATION (THIS SECTION IS 7784
!   TERMINATED WHEN TCONTINUE2 = "N") 7785
! 3. IF TCONTINUE1 = "N", SECTION TO REQUEST RESPONSE TO CONTINUE 7786
!   PROCEDURE OR EXIT TO MENU 7787
! 7788
! ===== 7789
! 7790
! TCONTINUE1 = "Y" 7791
! WHILE TCONTINUE1 = "Y" 7792
! BEGIN 7793
! 7794
! ===== 7795
! 7796
! LOOP TO DISPLAY BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE 7797
! FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 7798
! 7799
! ===== 7800
! 7801
! 7802
! 7803
! FLG1 = "N" 7804
! IMSG = 1 7805
! WHILE FLG1 = "N" 7806
! BEGIN 7807
!   IF IMSG = 1 THEN TMSG = TMSG1 7808
!   IF IMSG = 2 THEN TMSG = TMSG2 7809
!   IF IMSG = 3 THEN TMSG = TMSG3 7810
!   IF IMSG = 4 THEN TMSG = TMSG4 7811
!   IF IMSG = 5 THEN TMSG = TMSG5 7812
!   IF IMSG = 6 THEN TMSG = TMSG6 7813
!   IF IMSG = 7 THEN TMSG = TMSG7 || " " | TDAY | " " | TMSG7A || 7814
!   " " | TMONTH VIA MONTH_TABLE 7815
!   IF IMSG = 8 THEN TMSG = TMSG8 7816
!   IF IMSG = 9 THEN TMSG = TMSG9 7817
!   IF IMSG = 10 THEN TMSG = TMSG10 7818
! 7819
! ===== 7820
! 7821
! THIS SECTION DISPLAYS THE FIND REFERENCES FORM AND RETRIEVES 7822
! THE DATA ENTERED ON THE FORM 7823
! 7824
! ===== 7825
! 7826

```

Datatrieve Procedure REF_MODIFY (cont.)

FOR FIRST 1 REFERENCES_FORM	7827
BEGIN	7828
DISPLAY FORM REFERENCES FIN FORM IN	7829
DEV\$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING	7830
BEGIN	7831
IF IMSG NE 1 THEN	7832
BEGIN	7833
PUT_FORM AUTHOR_1 = TAUTOR1	7834
PUT_FORM AUTHOR_2 = TAUTOR2	7835
PUT_FORM AUTHOR_3 = TAUTOR3	7836
PUT_FORM AUTHOR_4 = TAUTOR4	7837
PUT_FORM TITLE_1 = TTITLE1	7838
PUT_FORM TITLE_2 = TTITLE2	7839
PUT_FORM ORGANIZATION = TSOURCE	7840
PUT_FORM DOCUMENT_NO = TDOCNO	7841
PUT_FORM DATE = TDOCDATE	7842
PUT_FORM CONTRACT_NO = TCONTNO	7843
END	7844
PUT_FORM CONTINUE = TCONTINUE1	7845
PUT_FORM MESSAGE = TMSG	7846
END RETRIEVE USING	7847
BEGIN	7848
TAUTOR1 = GET_FORM AUTHOR_1	7849
TAUTOR2 = GET_FORM AUTHOR_2	7850
TAUTOR3 = GET_FORM AUTHOR_3	7851
TAUTOR4 = GET_FORM AUTHOR_4	7852
TTITLE1 = GET_FORM TITLE_1	7853
TTITLE2 = GET_FORM TITLE_2	7854
TSOURCE = GET_FORM ORGANIZATION	7855
TDOCNO = GET_FORM DOCUMENT_NO	7856
TDOCDATE = GET_FORM DATE	7857
TCONTNO = GET_FORM CONTRACT_NO	7858
TCONTINUE1 = GET_FORM CONTINUE	7859
END	7860
END	7861
TDAY = FN\$STR_EXTRACT(TDOCDATE, 1, 2)	7862
TDAY1 = FN\$STR_EXTRACT(TDAY, 1, 1)	7863
TDAY2 = FN\$STR_EXTRACT(TDAY, 2, 1)	7864
TMONTH = FN\$STR_EXTRACT(TDOCDATE, 3, 3)	7865
TYEAR = FN\$STR_EXTRACT(TDOCDATE, 6, 4)	7866
TYEAR1 = FN\$STR_EXTRACT(TYEAR, 1, 1)	7867
TYEAR2 = FN\$STR_EXTRACT(TYEAR, 2, 1)	7868
TYEAR3 = FN\$STR_EXTRACT(TYEAR, 3, 1)	7869
TYEAR4 = FN\$STR_EXTRACT(TYEAR, 4, 1)	7870
=====	7871
IF TCONTINUE1 IS NOT EQUAL TO "N", ANY LEADING BLANKS WHICH	7872
WERE INADVERTENTLY ENTERED IN TAUTOR1, TAUTOR2, TAUTOR3,	7873
	7874
	7875
	7876
	7877

Datatrieve Procedure REF_MODIFY (cont.)

```

!      TAUTHOR4, TTITLE1, TTITLE2, TSOURCE, TDOCNO AND TCONTNO ARE 7878
!      REMOVED 7879
!      ===== 7880
!      7881
!      7882
IF TCONTINUE1 NE "N" THEN 7883
  BEGIN 7884
    I = 1 7885
    WHILE I LE 9 7886
      BEGIN 7887
        IF I = 1 THEN TEMP1 = TAUTOR1 7888
        IF I = 2 THEN TEMP1 = TAUTOR2 7889
        IF I = 3 THEN TEMP1 = TAUTOR3 7890
        IF I = 4 THEN TEMP1 = TAUTOR4 7891
        IF I = 5 THEN TEMP1 = TTITLE1 7892
        IF I = 6 THEN TEMP1 = TTITLE2 7893
        IF I = 7 THEN TEMP1 = TSOURCE 7894
        IF I = 8 THEN TEMP1 = TDOCNO 7895
        IF I = 9 THEN TEMP1 = TCONTNO 7896
        IF TEMP1 NE " " THEN 7897
          BEGIN 7898
            J = 1 7899
            WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " " 7900
              BEGIN 7901
                J = J + 1 7902
              END 7903
              TEMP1 = FN$STR_EXTRACT(TEMP1, J, 112 - J + 1) 7904
            END 7905
            IF I = 1 THEN TAUTOR1 = TEMP1 7906
            IF I = 2 THEN TAUTOR2 = TEMP1 7907
            IF I = 3 THEN TAUTOR3 = TEMP1 7908
            IF I = 4 THEN TAUTOR4 = TEMP1 7909
            IF I = 5 THEN TTITLE1 = TEMP1 7910
            IF I = 6 THEN TTITLE2 = TEMP1 7911
            IF I = 7 THEN TSOURCE = TEMP1 7912
            IF I = 8 THEN TDOCNO = TEMP1 7913
            IF I = 9 THEN TCONTNO = TEMP1 7914
            I = I + 1 7915
          END 7916
        END 7917
      END 7918
    END 7919
  END 7920
  ===== 7921
  IF TCONTINUE1 IS NOT EQUAL TO "N", TESTS ARE PERFORMED TO 7922
  VERIFY ANY DATA ENTERED ON THE FORM. AS APPROPRIATE, THE 7923
  PROCEDURE CHECKS ANY OR ALL OF THE FOLLOWING: 7924
    1. DATA HAS BEEN ENTERED IN AT LEAST ONE OF THE FIELDS 7925
    2. TTITLE1 DOES NOT END WITH A HYPENATED WORD 7926
    3. TSOURCE IS IN REFERENCE SOURCE TABLE 7927
    4. TDAY IS NUMERIC IN RANGE 0 TO 31 INCLUSIVE 7928

```

Datatrieve Procedure REF_MODIFY (cont.)

5. TMONTH IS IN MONTH TABLE	7929
6. NDAY IS LESS THAN OR EQUAL TO THE NUMBER OF DAYS IN TMONTH	7930
7. TYEAR IS NUMERIC IN RANGE 1960 TO 9999 INCLUSIVE	7931
8. TCONTINUE1 IS "Y" OR "N"	7932
9. AT LEAST ONE RECORD EXISTS WITH THE DATA SPECIFIED	7933
=====	7934
	7935
	7936
FLG1 = "Y"	7937
IF TCONTINUE1 NE "N" THEN	7938
BEGIN	7939
IF TAUTHOR1 = " " AND TAUTHOR2 = " " AND	7940
TAUTHOR3 = " " AND TAUTHOR4 = " " AND	7941
TTITLE1 = " " AND TTITLE2 = " " AND	7942
TDOCDATE = " " AND TSOURCE = " " AND	7943
TDOCNO = " " AND TCONTNO = " " THEN	7944
BEGIN	7945
FLG1 = "N"	7946
IMSG = 2	7947
END	7948
IF FLG1 = "Y" AND	7949
TTITLE1 NE " " THEN	7950
BEGIN	7951
J = 80	7952
WHILE FN\$STR_EXTRACT(TTITLE1, J, 1) = " "	7953
BEGIN	7954
J = J - 1	7955
END	7956
IF FN\$STR_EXTRACT(TTITLE1, J, 1) = "-" AND	7957
FN\$STR_EXTRACT(TTITLE1, J - 1, 1) NE " " AND	7958
FN\$STR_EXTRACT(TTITLE1, J - 2, 2) NE " -" THEN	7959
BEGIN	7960
FLG1 = "N"	7961
IMSG = 3	7962
END	7963
END	7964
END	7965
IF FLG1 = "Y" AND	7966
TSOURCE NE " " THEN	7967
BEGIN	7968
IF TSOURCE NOT IN REFERENCE_SOURCE_TABLE THEN	7969
BEGIN	7970
FLG1 = "N"	7971
IMSG = 4	7972
END	7973
END	7974
IF FLG1 = "Y" AND	7975
TDOCDATE NE " " THEN	7976
BEGIN	7977
FLG2 = "N"	7978
IF (TDAY1 = " " OR TDAY1 IN NUMBER_TABLE) AND	7979

Datatrieve Procedure REF_MODIFY (cont.)

```

(TDAY2 = " " OR TDAY2 IN NUMBER_TABLE) THEN      7980
BEGIN                                              7981
    NDAY = TDAY                                  7982
    IF NDAY GE 0 AND                              7983
        NDAY LE 31 THEN                          7984
        FLG2 = "Y"                               7985
    END                                           7986
IF FLG2 = "Y" THEN                               7987
    BEGIN                                        7988
        TDAY = NDAY                             7989
        IF NDAY LT 10 THEN TDAY = "0" | TDAY     7990
    END                                           7991
IF FLG2 = "N" THEN                               7992
    BEGIN                                        7993
        FLG1 = "N"                              7994
        IMMSG = 5                                7995
    END                                           7996
END                                               7997
IF FLG1 = "Y" AND                                7998
    TDOCDATE NE " " THEN                        7999
    BEGIN                                        8000
        IF TMONTH NOT IN MONTH_TABLE THEN      8001
        BEGIN                                  8002
            FLG1 = "N"                         8003
            IMMSG = 6                          8004
        END                                    8005
    END                                           8006
IF FLG1 = "Y" AND                                8007
    TDOCDATE NE " " THEN                        8008
    BEGIN                                        8009
        IF (TMONTH = "JAN" AND NDAY GT 31) OR  8010
            (TMONTH = "FEB" AND NDAY GT 29) OR  8011
            (TMONTH = "MAR" AND NDAY GT 31) OR  8012
            (TMONTH = "APR" AND NDAY GT 30) OR  8013
            (TMONTH = "MAY" AND NDAY GT 31) OR  8014
            (TMONTH = "JUN" AND NDAY GT 30) OR  8015
            (TMONTH = "JUL" AND NDAY GT 31) OR  8016
            (TMONTH = "AUG" AND NDAY GT 31) OR  8017
            (TMONTH = "SEP" AND NDAY GT 30) OR  8018
            (TMONTH = "OCT" AND NDAY GT 31) OR  8019
            (TMONTH = "NOV" AND NDAY GT 30) OR  8020
            (TMONTH = "DEC" AND NDAY GT 31) THEN 8021
        BEGIN                                  8022
            FLG1 = "N"                         8023
            IMMSG = 7                          8024
        END                                    8025
    END                                           8026
IF FLG1 = "Y" AND                                8027
    TDOCDATE NE " " THEN                        8028
    BEGIN                                        8029
        FLG2 = "N"                            8030

```

Datatrieve Procedure REF_MODIFY (cont.)

```

      IF (TYEAR1 = " " OR TYEAR1 IN NUMBER_TABLE) AND      8031
        (TYEAR2 = " " OR TYEAR2 IN NUMBER_TABLE) AND      8032
        (TYEAR3 = " " OR TYEAR3 IN NUMBER_TABLE) AND      8033
        (TYEAR4 = " " OR TYEAR4 IN NUMBER_TABLE) THEN      8034
      BEGIN                                                  8035
        NYEAR = TYEAR                                       8036
        IF NYEAR GE 1960 AND                                8037
          NYEAR LE 9999 THEN                                8038
          FLG2 = "Y"                                         8039
        END                                                  8040
      IF FLG2 = "Y" THEN                                     8041
        TDOCDATE = TDAY | TMONTH | TYEAR                    8042
      IF FLG2 = "N" THEN                                     8043
      BEGIN                                                  8044
        FLG1 = "N"                                           8045
        IMSG = 8                                              8046
      END                                                    8047
    END                                                      8048
  IF FLG1 = "Y" THEN                                        8049
  BEGIN                                                      8050
    IF TCONTINUE1 NE "Y" AND                                8051
      TCONTINUE1 NE "N" THEN                                8052
    BEGIN                                                    8053
      FLG1 = "N"                                             8054
      IMSG = 9                                               8055
    END                                                      8056
  END                                                        8057
IF FLG1 = "Y" THEN                                        8058
BEGIN                                                      8059
  TTITLE = TTITLE1                                          8060
  IF TTITLE = " " THEN                                      8061
    TTITLE = TTITLE2 ELSE                                    8062
    TTITLE = TTITLE || " " | TTITLE2                        8063
  END                                                        8064
IF FLG1 = "Y" THEN                                        8065
BEGIN                                                      8066
  FLG2 = "N"                                                8067
  IF TTITLE NE " " AND TSOURCE NE " " THEN                 8068
  BEGIN                                                    8069
    FLG2 = "Y"                                              8070
    JCNT = 0                                                8071
    FOR REFERENCES WITH                                     8072
      DOCUMENT_TITLE = TTITLE AND                           8073
      DOCUMENT_SOURCE = TSOURCE                             8074
    BEGIN                                                  8075
      FLG3 = "Y"                                           8076
      IF TAUTHOR1 NE " " AND                                8077
        TAUTHOR1 NE AUTHOR1 THEN FLG3 = "N"                8078
      IF TAUTHOR2 NE " " AND                                8079
        TAUTHOR2 NE AUTHOR2 THEN FLG3 = "N"                8080
      IF TAUTHOR3 NE " " AND                                8081

```


Datatrieve Procedure REF_MODIFY (cont.)

```

            TAUTHOR3 NE AUTHOR3 THEN FLG3 = "N"      8082
        IF TAUTHOR4 NE " " AND                        8083
            TAUTHOR4 NE AUTHOR4 THEN FLG3 = "N"      8084
        IF TDOCNO NE " " AND                          8085
            TDOCNO NE DOCUMENT_NUMBER THEN          8086
            FLG3 = "N"                                8087
        IF TDOCDATE NE " " AND                        8088
            TDAY | "-" |                             8089
            TMONTH | "-" |                            8090
            TYEAR NE DOCUMENT_DATE THEN             8091
            FLG3 = "N"                                8092
        IF TCONTNO NE " " AND                        8093
            TCONTNO NE CONTRACT_NUMBER THEN         8094
            FLG3 = "N"                                8095
        IF FLG3 = "Y" THEN JCNT = JCNT + 1          8096
    END                                              8097
END                                              8098
IF FLG2 = "N" THEN                                8099
    BEGIN                                          8100
        JCNT = 0                                  8101
        FOR REFERENCES                            8102
        BEGIN                                      8103
            FLG3 = "Y"                             8104
            IF TAUTHOR1 NE " " AND                 8105
                TAUTHOR1 NE AUTHOR1 THEN FLG3 = "N" 8106
            IF TAUTHOR2 NE " " AND                 8107
                TAUTHOR2 NE AUTHOR2 THEN FLG3 = "N" 8108
            IF TAUTHOR3 NE " " AND                 8109
                TAUTHOR3 NE AUTHOR3 THEN FLG3 = "N" 8110
            IF TAUTHOR4 NE " " AND                 8111
                TAUTHOR4 NE AUTHOR4 THEN FLG3 = "N" 8112
            IF TTITLE NE " " AND                   8113
                TTITLE NE DOCUMENT_TITLE THEN      8114
                FLG3 = "N"                          8115
            IF TSOURCE NE " " AND                  8116
                TSOURCE NE DOCUMENT_SOURCE THEN    8117
                FLG3 = "N"                          8118
            IF TDOCNO NE " " AND                   8119
                TDOCNO NE DOCUMENT_NUMBER THEN     8120
                FLG3 = "N"                          8121
            IF TDOCDATE NE " " AND                 8122
                TDAY | "-" |                       8123
                TMONTH | "-" |                     8124
                TYEAR NE DOCUMENT_DATE THEN        8125
                FLG3 = "N"                          8126
            IF TCONTNO NE " " AND                  8127
                TCONTNO NE CONTRACT_NUMBER THEN    8128
                FLG3 = "N"                          8129
            IF FLG3 = "Y" THEN JCNT = JCNT + 1      8130
        END                                          8131
    END                                              8132

```


Datatrieve Procedure REF_MODIFY (cont.)

FLG3 = "N"	8184
IF TCONTNO NE " " AND	8185
TCONTNO NE CONTRACT_NUMBER THEN	8186
FLG3 = "N"	8187
IF FLG3 = "Y" THEN	8188
BEGIN	8189
ICNT = ICNT + 1	8190
:REF_MODIFY_1	8191
END	8192
END	8193
END	8194
IF FLG2 = "N" THEN	8195
BEGIN	8196
ICNT = 0	8197
FOR REFERENCES SORTED BY REFERENCE_NUMBER	8198
BEGIN	8199
IF TCONTINUE2 = "Y" THEN	8200
BEGIN	8201
FLG3 = "Y"	8202
IF TAUTHOR1 NE " " AND	8203
TAUTHOR1 NE AUTHOR1 THEN FLG3 = "N"	8204
IF TAUTHOR2 NE " " AND	8205
TAUTHOR2 NE AUTHOR2 THEN FLG3 = "N"	8206
IF TAUTHOR3 NE " " AND	8207
TAUTHOR3 NE AUTHOR3 THEN FLG3 = "N"	8208
IF TAUTHOR4 NE " " AND	8209
TAUTHOR4 NE AUTHOR4 THEN FLG3 = "N"	8210
IF TTITLE NE " " AND	8211
TTITLE NE DOCUMENT_TITLE THEN	8212
FLG3 = "N"	8213
IF TSOURCE NE " " AND	8214
TSOURCE NE DOCUMENT_SOURCE THEN	8215
FLG3 = "N"	8216
IF TDOCNO NE " " AND	8217
TDOCNO NE DOCUMENT_NUMBER THEN	8218
FLG3 = "N"	8219
IF TDOCDATE NE " " AND	8220
TDAY "-"	8221
TMONTH "-"	8222
TYEAR NE DOCUMENT_DATE THEN	8223
FLG3 = "N"	8224
IF TCONTNO NE " " AND	8225
TCONTNO NE CONTRACT_NUMBER THEN	8226
FLG3 = "N"	8227
IF FLG3 = "Y" THEN	8228
BEGIN	8229
ICNT = ICNT + 1	8230
:REF_MODIFY_1	8231
END	8232
END	8233
END	8234

Datatrieve Procedure REF_MODIFY (cont.)

END	8235
END	8236
END	8237
=====	8238
=====	8239
=====	8240
IF TCONTINUE1 IS EQUAL TO "N", A RESPONSE IS REQUESTED TO EITHER	8241
CONTINUE THE PROCEDURE TO MODIFY REFERENCES OR EXIT TO THE MENU	8242
=====	8243
=====	8244
=====	8245
IF TCONTINUE1 = "N" THEN	8246
BEGIN	8247
PRINT NEW_PAGE	8248
:CLRSCRN	8249
TCONTINUE1 = "X"	8250
WHILE TCONTINUE1 NE "Y" AND	8251
TCONTINUE1 NE "N"	8252
BEGIN	8253
PRINT SKIP 2,	8254
"Do you wish to continue modifying", SKIP 1,	8255
"REFERENCES?", SKIP 1	8256
TCONTINUE1 = FN\$UPCASE(*."Y or N")	8257
PRINT " "	8258
END	8259
END	8260
END	8261
END-PROCEDURE	8262
	8263

Datatrieve Procedure REF_MODIFY_1

DEFINE PROCEDURE REF_MODIFY_1	8264
!	8265
!	8266
=====	8267
!	8268
! THE FIELDS OF THE INCOMING REFERENCE RECORD ARE ASSIGNED TO VARIABLES	8269
! FOR DISPLAY AND MODIFICATION. THE FIELDS DOCUMENT_TITLE,	8270
! DOCUMENT_SOURCE AND DOCUMENT_DATE ARE NOT ASSIGNED TO VARIABLES SINCE	8271
! THESE FIELDS CANNOT BE MODIFIED.	8272
!	8273
=====	8274
!	8275
TTAUTHOR1 = AUTHOR1	8276
TTAUTHOR2 = AUTHOR2	8277
TTAUTHOR3 = AUTHOR3	8278
TTAUTHOR4 = AUTHOR4	8279
TTDOCNO = DOCUMENT_NUMBER	8280
TTCONTNO = CONTRACT_NUMBER	8281
!	8282
!	8283
=====	8284
!	8285
! THIS SECTION SPLITS THE FIELD DOCUMENT_TITLE INTO VARIABLES WHICH ARE	8286
! 80 CHARACTERS LONG FOR DISPLAY ON THE TDMS FORM	8287
!	8288
=====	8289
!	8290
XTITLE = DOCUMENT_TITLE	8291
I = 1	8292
WHILE I LE 3	8293
BEGIN	8294
IF XTITLE NE " " THEN	8295
BEGIN	8296
J = 1	8297
WHILE FN\$STR_EXTRACT(XTITLE, J, 1) = " "	8298
BEGIN	8299
J = J + 1	8300
END	8301
XTITLE = FN\$STR_EXTRACT(XTITLE, J, 242 - J + 1)	8302
END	8303
FLG4 = "N"	8304
IF FN\$STR_EXTRACT(XTITLE, 80, 1) = " " OR	8305
FN\$STR_EXTRACT(XTITLE, 81, 1) = " " THEN	8306
BEGIN	8307
TEMP2 = FN\$STR_EXTRACT(XTITLE, 1, 80)	8308
XTITLE = FN\$STR_EXTRACT(XTITLE, 81, 162)	8309
FLG4 = "Y"	8310
END	8311
IF FLG4 = "N" THEN	8312
BEGIN	8313
J = 80	8314

Datatrieve Procedure REF_MODIFY_1 (cont.)

```

      WHILE FN$STR_EXTRACT(XTITLE, J, 1) NE " " AND
        J GT 1
      BEGIN
        J = J - 1
      END
    CHOICE
      J = 1 THEN TEMP2 = FN$STR_EXTRACT(XTITLE, 1, 80)
      J GT 1 THEN TEMP2 = FN$STR_EXTRACT(XTITLE, 1, J - 1 + 1)
    END_CHOICE
  CHOICE
    J = 1 THEN XTITLE = FN$STR_EXTRACT(XTITLE, 81, 162)
    J GT 1 THEN XTITLE = FN$STR_EXTRACT(XTITLE, J + 1,
      242 - (J + 1) + 1)
  END_CHOICE
END
IF I = 1 THEN XTITLE1 = TEMP2
IF I = 2 THEN XTITLE2 = TEMP2
I = I + 1
END

=====
!
! LOOP TO DISPLAY A REFERENCE RECORD USING A TDMS FORM, RETRIEVE DATA
! FROM THE FORM, TEST THE INCOMING INFORMATION AND REQUEST CORRECTION OF
! INVALID DATA
!
=====
!
FLG4 = "N"
IMSG = 11
WHILE FLG4 = "N"
  BEGIN
    IF IMSG = 11 THEN TMSG = TMSG11
    IF IMSG = 12 THEN TMSG = TMSG12

    =====
    !
    ! THIS SECTION DISPLAYS THE MODIFY REFERENCES FORM AND RETRIEVES
    ! THE DATA ENTERED ON THE FORM
    !
    =====
  END

  DISPLAY FORM REFERENCES MOD FORM IN
  DEV$Z06:[BCDSSME2.FORMS]FORMSLIB.RLB USING
  BEGIN
    PUT_FORM RECORD_NUMBER = ICNT
    PUT_FORM TOTAL_RECORDS = JCNT
    PUT_FORM AUTHOR_1      = TTAUTHOR1
    PUT_FORM AUTHOR_2      = TTAUTHOR2

```

Datatrieve Procedure REF_MODIFY_1 (cont.)

```

      PUT_FORM AUTHOR_3      = TTAUTHOR3      8366
      PUT_FORM AUTHOR_4      = TTAUTHOR4      8367
      PUT_FORM TITLE_1       = XTITLE1        8368
      PUT_FORM TITLE_2       = XTITLE2        8369
      PUT_FORM ORGANIZATION  = DOCUMENT_SOURCE 8370
      PUT_FORM DOCUMENT_NO   = TTDOCNO        8371
      PUT_FORM DATE          = DOCUMENT_DATE   8372
      PUT_FORM CONTRACT_NO   = TTCONTNO       8373
      IF IMSG = 11 THEN      8374
        PUT_FORM CONTINUE    = "Y"           8375
      IF IMSG NE 11 THEN    8376
        PUT_FORM CONTINUE    = TCONTINUE2     8377
      PUT_FORM MESSAGE       = TMSG           8378
END RETRIEVE USING        8379
BEGIN                     8380
  TTAUTHOR1 = GET_FORM AUTHOR_1             8381
  TTAUTHOR2 = GET_FORM AUTHOR_2             8382
  TTAUTHOR3 = GET_FORM AUTHOR_3             8383
  TTAUTHOR4 = GET_FORM AUTHOR_4             8384
  TTDOCNO   = GET_FORM DOCUMENT_NO          8385
  TTCONTNO  = GET_FORM CONTRACT_NO          8386
  TCONTINUE2 = GET_FORM CONTINUE            8387
END                                           8388
!                                           8389
!                                           8390
!                                           8391
!===== 8392
! IF TCONTINUE2 IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH WERE
! INADVERTENTLY ENTERED IN TTAUTHOR1, TTAUTHOR2, TTAUTHOR3,
! TTAUTHOR4, TTDOCNO AND TTCONTNO ARE REMOVED 8393
!===== 8394
!                                           8395
!                                           8396
!                                           8397
!                                           8398
! IF TCONTINUE2 NE "A" THEN 8399
! BEGIN                     8400
!   I = 1                   8401
!   WHILE I LE 6            8402
!     BEGIN                 8403
!       IF I = 1 THEN TEMP1 = TTAUTHOR1    8404
!       IF I = 2 THEN TEMP1 = TTAUTHOR2    8405
!       IF I = 3 THEN TEMP1 = TTAUTHOR3    8406
!       IF I = 4 THEN TEMP1 = TTAUTHOR4    8407
!       IF I = 5 THEN TEMP1 = TTDOCNO      8408
!       IF I = 6 THEN TEMP1 = TTCONTNO     8409
!       IF TEMP1 NE " " THEN              8410
!         BEGIN                     8411
!           J = 1                   8412
!           WHILE FN$STR_EXTRACT(TEMP1, J, 1) = " " 8413
!             BEGIN                 8414
!               J = J + 1            8415
!             END                     8416

```


Datatrieve Procedure REF_MODIFY_1 (cont.)

```

J = 1                                                    8468
WHILE J = 1                                              8469
  BEGIN                                                  8470
    J = FN$STR_LOC(TEMP1, "$$")                          8471
    IF J = 0 THEN TAUTHOR = " "                          8472
    IF J = 1 THEN TEMP1 = FN$STR_EXTRACT(TEMP1, 4, 109)  8473
    IF J GT 1 THEN                                       8474
      BEGIN                                              8475
        TAUTHOR = FN$STR_EXTRACT(TEMP1, 1, J - 1)        8476
        TEMP1 = FN$STR_EXTRACT(TEMP1, J, 112 - J + 1)    8477
      END                                                  8478
    END                                                  8479
    IF I = 1 THEN TTAUTHOR1 = TAUTHOR                    8480
    IF I = 2 THEN TTAUTHOR2 = TAUTHOR                    8481
    IF I = 3 THEN TTAUTHOR3 = TAUTHOR                    8482
    IF I = 4 THEN TTAUTHOR4 = TAUTHOR                    8483
    I = I + 1                                             8484
  END                                                    8485
END                                                       8486
!                                                       8487
!                                                       8488
! ===== 8489
! IF TCONTINUE2 IS NOT EQUAL TO "A" AND NEW DATA HAS BEEN ENTERED IN AT 8490
! LEAST ONE OF THE RECORD FIELDS, THIS SECTION PRINTS THE INITIAL RECORD 8491
! DATA TO THE LOG FILE, MODIFIES THE RECORD AND PRINTS THE MODIFIED      8492
! RECORD DATA TO THE LOG FILE (HIGHLIGHTING THE CHANGED FIELDS)          8493
! ===== 8494
!                                                       8495
! IF TCONTINUE2 NE "A" AND 8496
! (AUTHOR1 NE TTAUTHOR1 OR 8497
!  AUTHOR2 NE TTAUTHOR2 OR 8498
!  AUTHOR3 NE TTAUTHOR3 OR 8499
!  AUTHOR4 NE TTAUTHOR4 OR 8500
!  DOCUMENT_NUMBER NE TTDOCNO OR 8501
!  CONTRACT_NUMBER NE TTCONTNO) THEN 8502
! BEGIN 8503
!                                           8504
! ===== 8505
! THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG 8506
! FILE 8507
! ===== 8508
!                                           8509
!                                           8510
!                                           8511
!                                           8512
!                                           8513
!                                           8514
! KCNT = KCNT + 1 8515
! PRINT NEW_PAGE, SKIP 3, 8516
!   COL 1, "RECORD NO. ", SPACE 0, 8517
!     KCNT (-) USING 9(4), SKIP 1, 8518

```

Datatrieve Procedure REF_MODIFY_1 (cont.)

```

=====, SPACE 0, 8519
=====, SKIP 1, 8520
=====, SPACE 0, 8521
=====, SKIP 2, 8522
COL 6, "DATE_CREATED      :", SPACE 1, 8523
      DATE_CREATED (-) USING X(23), SKIP 1, 8524
COL 6, "REFERENCE_NUMBER  :", SPACE 1, 8525
      REFERENCE_NUMBER (-) USING X(5), SKIP 1, 8526
COL 6, "AUTHOR1           :", SPACE 1, 8527
      AUTHOR1 (-) USING X(25), SKIP 1, 8528
COL 6, "AUTHOR2           :", SPACE 1, 8529
      AUTHOR2 (-) USING X(25), SKIP 1, 8530
COL 6, "AUTHOR3           :", SPACE 1, 8531
      AUTHOR3 (-) USING X(25), SKIP 1, 8532
COL 6, "AUTHOR4           :", SPACE 1, 8533
      AUTHOR4 (-) USING X(25), SKIP 1, 8534
COL 6, "DOCUMENT_TITLE    :", SPACE 1, 8535
      DOCUMENT_TITLE (-) USING T(52), SKIP 1, 8536
COL 6, "DOCUMENT_SOURCE   :", SPACE 1, 8537
      DOCUMENT_SOURCE (-) USING X(30), SKIP 1, 8538
COL 6, "DOCUMENT_NUMBER   :", SPACE 1, 8539
      DOCUMENT_NUMBER (-) USING X(30), SKIP 1, 8540
COL 6, "DOCUMENT_DATE     :", SPACE 1, 8541
      DOCUMENT_DATE (-) USING X(11), SKIP 1, 8542
COL 6, "CONTRACT_NUMBER   :", SPACE 1, 8543
      CONTRACT_NUMBER (-) USING X(20), SKIP 1, 8544
COL 6, "DATE_LAST_MODIFIED :", SPACE 1, 8545
      DATE_LAST_MODIFIED (-) USING X(23), SKIP 1, 8546
COL 6, "MODIFYING_PROCEDURE :", SPACE 1, 8547
      MODIFYING_PROCEDURE (-) USING X(20) 8548

```

```

=====
THIS SECTION STORES THE INITIAL RECORD VALUES FOR USE IN
HIGHLIGHTING THE FIELDS WHICH HAVE BEEN MODIFIED
=====

```

```

HDCREATED = DATE_CREATED 8558
HREFNO    = REFERENCE_NUMBER 8559
HAUTHOR1  = AUTHOR1 8560
HAUTHOR2  = AUTHOR2 8561
HAUTHOR3  = AUTHOR3 8562
HAUTHOR4  = AUTHOR4 8563
HTITLE    = DOCUMENT_TITLE 8564
HSOURCE   = DOCUMENT_SOURCE 8565
HDOCNO    = DOCUMENT_NUMBER 8566
HDOCDATE  = DOCUMENT_DATE 8567
HCONTNO   = CONTRACT_NUMBER 8568
HDLASTMOD = DATE_LAST_MODIFIED 8569

```

Datatrieve Procedure REF_MODIFY_1 (cont.)

```

HMODPROC = MODIFYING_PROCEDURE                                8570
                                                                8571
                                                                8572
=====                                                       8573
                                                                8574
THIS SECTION MODIFIES THE REFERENCE RECORD USING THE VERIFIED  8575
DATA ENTERED ON THE MODIFY REFERENCES FORM                     8576
                                                                8577
=====                                                       8578
                                                                8579
CAL = "NOW"                                                    8580
MODIFY USING                                                    8581
BEGIN                                                         8582
    AUTHOR1 = TTAUTHOR1                                         8583
    AUTHOR2 = TTAUTHOR2                                         8584
    AUTHOR3 = TTAUTHOR3                                         8585
    AUTHOR4 = TTAUTHOR4                                         8586
    DOCUMENT_NUMBER = TTDOCNO                                   8587
    CONTRACT_NUMBER = TTCONTNO                                  8588
    DATE_LAST_MODIFIED = CAL                                    8589
    MODIFYING_PROCEDURE = "REF_MODIFY"                         8590
END                                                            8591
                                                                8592
                                                                8593
=====                                                       8594
                                                                8595
THIS SECTION PRINTS THE MODIFIED RECORD DATA TO THE SESSION LOG 8596
FILE AND HIGHLIGHTS THE FIELDS WHICH CONTAIN NEW INFORMATION 8597
                                                                8598
=====                                                       8599
                                                                8600
PRINT SKIP 1,                                                  8601
COL 1, "===== ", SPACE 0,                                     8602
      "===== ", SKIP 2,                                       8603
COL 1, CHOICE                                                  8604
    DATE_CREATED = HDCREATED THEN " "                          8605
    ELSE "***"                                                  8606
END CHOICE,                                                    8607
SPACE 2, "DATE_CREATED :", SPACE 1,                            8608
      DATE_CREATED (-) USING X(23), SKIP 1,                     8609
COL 1, CHOICE                                                  8610
    REFERENCE_NUMBER = HREFNO THEN " "                         8611
    ELSE "***"                                                  8612
END CHOICE,                                                    8613
SPACE 2, "REFERENCE_NUMBER :", SPACE 1,                        8614
      REFERENCE_NUMBER (-) USING X(5), SKIP 1,                 8615
COL 1, CHOICE                                                  8616
    AUTHOR1 = HAUTHOR1 THEN " "                                8617
    ELSE "***"                                                  8618
END CHOICE,                                                    8619
SPACE 2, "AUTHOR1 :", SPACE 1,                                 8620

```

Datatrieve Procedure REF_MODIFY_1 (cont.)

	AUTHOR1 (-) USING X(25), SKIP 1,	8621
COL 1, CHOICE		8622
	AUTHOR2 = HAUTHOR2 THEN " "	8623
	ELSE "***"	8624
	END CHOICE,	8625
	SPACE 2, "AUTHOR2 : ", SPACE 1,	8626
	AUTHOR2 (-) USING X(25), SKIP 1,	8627
COL 1, CHOICE		8628
	AUTHOR3 = HAUTHOR3 THEN " "	8629
	ELSE "***"	8630
	END CHOICE,	8631
	SPACE 2, "AUTHOR3 : ", SPACE 1,	8632
	AUTHOR3 (-) USING X(25), SKIP 1,	8633
COL 1, CHOICE		8634
	AUTHOR4 = HAUTHOR4 THEN " "	8635
	ELSE "***"	8636
	END CHOICE,	8637
	SPACE 2, "AUTHOR4 : ", SPACE 1,	8638
	AUTHOR4 (-) USING X(25), SKIP 1,	8639
COL 1, CHOICE		8640
	DOCUMENT_TITLE = HTITLE THEN " "	8641
	ELSE "***"	8642
	END CHOICE,	8643
	SPACE 2, "DOCUMENT_TITLE : ", SPACE 1,	8644
	DOCUMENT_TITLE (-) USING T(52), SKIP 1,	8645
COL 1, CHOICE		8646
	DOCUMENT_SOURCE = HSOURCE THEN " "	8647
	ELSE "***"	8648
	END CHOICE,	8649
	SPACE 2, "DOCUMENT_SOURCE : ", SPACE 1,	8650
	DOCUMENT_SOURCE (-) USING X(30), SKIP 1,	8651
COL 1, CHOICE		8652
	DOCUMENT_NUMBER = HDOCNO THEN " "	8653
	ELSE "***"	8654
	END CHOICE,	8655
	SPACE 2, "DOCUMENT_NUMBER : ", SPACE 1,	8656
	DOCUMENT_NUMBER (-) USING X(30), SKIP 1,	8657
COL 1, CHOICE		8658
	DOCUMENT_DATE = HDOCDATE THEN " "	8659
	ELSE "***"	8660
	END CHOICE,	8661
	SPACE 2, "DOCUMENT_DATE : ", SPACE 1,	8662
	DOCUMENT_DATE (-) USING X(11), SKIP 1,	8663
COL 1, CHOICE		8664
	CONTRACT_NUMBER = HCONTNO THEN " "	8665
	ELSE "***"	8666
	END CHOICE,	8667
	SPACE 2, "CONTRACT_NUMBER : ", SPACE 1,	8668
	CONTRACT_NUMBER (-) USING T(52), SKIP 1,	8669
COL 1, CHOICE		8670
	DATE_LAST_MODIFIED = HDLASTMOD THEN " "	8671

Datatrieve Procedure REF_MODIFY_1 (cont.)

```

                ELSE "****"
                END_CHOICE,
                SPACE 2, "DATE_LAST_MODIFIED :", SPACE 1,
                DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,
COL 1, CHOICE
                MODIFYING_PROCEDURE = HMODPROC THEN " "
                ELSE "****"
                END_CHOICE,
                SPACE 2, "MODIFYING_PROCEDURE :", SPACE 1,
                MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,
COL 1, "=====", SPACE 0,
        "=====", SKIP 1,
        "=====", SPACE 0,
        "=====
: BELL
END
!
!
!-----
! IF TCONTINUE2 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN MODIFIED. THIS
! MESSAGE WILL APPEAR IN THE SESSION LOG FILE.
!-----
!
IF TCONTINUE2 = "A" THEN
    BEGIN
        PRINT NEW_PAGE, SKIP 3,
        COL 1, "=====", SPACE 0,
        "=====", SKIP 10,
        COL 9, "*****", SKIP 2,
        COL 9, "***** RECORD NOT MODIFIED *****", SKIP 2,
        COL 9, "*****", SKIP 10,
        COL 1, "=====", SPACE 0,
        "=====
    END
END-PROCEDURE

```

Datatrieve Procedure REF_STORE

DEFINE PROCEDURE REF_STORE	8710
!	8711
!	8712
=====	8713
!	8714
! VARIABLES ASSOCIATED WITH AUTHORS	8715
!	8716
=====	8717
!	8718
DECLARE TAUTOR1 PIC X(25).	8719
DECLARE TAUTOR2 PIC X(25).	8720
DECLARE TAUTOR3 PIC X(25).	8721
DECLARE TAUTOR4 PIC X(25).	8722
DECLARE TTAUTOR PIC X(25).	8723
!	8724
=====	8725
!	8726
!	8727
! VARIABLES ASSOCIATED WITH DOCUMENT_TITLE	8728
!	8729
=====	8730
!	8731
DECLARE TTITLE1 PIC X(80).	8732
DECLARE TTITLE2 PIC X(80).	8733
DECLARE TTITLE PIC X(161).	8734
!	8735
!	8736
=====	8737
!	8738
! VARIABLES ASSOCIATED WITH DOCUMENT_DATE	8739
!	8740
=====	8741
!	8742
DECLARE TDOCDATE PIC X(9).	8743
DECLARE TDAY PIC X(2).	8744
DECLARE NDAY PIC 9(2).	8745
DECLARE TMONTH PIC X(3).	8746
DECLARE TYEAR PIC X(4).	8747
DECLARE NYEAR PIC 9(4).	8748
!	8749
=====	8750
!	8751
!	8752
! VARIABLES ASSOCIATED WITH OTHER INPUT FIELDS FOR	8753
! DOMAIN REFERENCES:	8754
! 1. SEQUENCE_NUMBER	8755
! 2. DOCUMENT_SOURCE	8756
! 3. DOCUMENT_NUMBER	8757
! 4. CONTRACT_NUMBER	8758
!	8759
=====	8760

Datatrieve Procedure REF_STORE (cont.)

!		8761
DECLARE TSEQNO	PIC 9(3).	8762
DECLARE TSOURCE	PIC X(30).	8763
DECLARE TDOCNO	PIC X(30).	8764
DECLARE TCONTNO	PIC X(20).	8765
!		8766
!		8767
=====		8768
!		8769
!	VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR AUTHORS,	8770
!	DOCUMENT_TITLE, DOCUMENT_SOURCE, DOCUMENT_NUMBER AND CONTRACT_NUMBER	8771
!		8772
=====		8773
!		8774
DECLARE TEMP	PIC X(112).	8775
!		8776
!		8777
=====		8778
!		8779
!	VARIABLES USED AS FLAGS OR CONDITION INDICATORS	8780
!		8781
=====		8782
!		8783
DECLARE TCONTINUE	PIC X(1).	8784
DECLARE MSG	PIC 9(2).	8785
DECLARE FLG1	PIC X(1).	8786
!		8787
!		8788
=====		8789
!		8790
!	VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM	8791
!		8792
=====		8793
!		8794
DECLARE TMSG	PIC X(80).	8795
DECLARE TMSG1	PIC X(80).	8796
DECLARE TMSG2	PIC X(80).	8797
DECLARE TMSG3	PIC X(80).	8798
DECLARE TMSG4	PIC X(80).	8799
DECLARE TMSG5	PIC X(80).	8800
DECLARE TMSG5A	PIC X(80).	8801
DECLARE TMSG6	PIC X(80).	8802
DECLARE TMSG7	PIC X(80).	8803
DECLARE TMSG8	PIC X(80).	8804
!		8805
=====		8806
!		8807
!	VARIABLES USED AS COUNTERS	8808
!		8809
=====		8810
!		8811

Datatrieve Procedure REF_STORE (cont.)

```

DECLARE ICNT      PIC 9(4).      8812
DECLARE JCNT      PIC 9(4).      8813
DECLARE I         PIC 9(4).      8814
DECLARE J         PIC 9(4).      8815
!                                                         8816
!                                                         8817
!===== 8818
! VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE 8819
!===== 8820
!                                                         8821
!===== 8822
DECLARE CAL        USAGE DATE    8823
                        EDIT_STRING X(23). 8824
!                                                         8825
!                                                         8826
!===== 8827
!                                                         8828
! READY THE APPROPRIATE DOMAINS, INITIALIZE THE MESSAGE VARIABLES, 8829
! INITIALIZE THE COUNTER (ICNT) USED FOR NUMBERING THE LOG FILE RECORDS 8830
! AND INITIALIZE THE COUNTER (JCNT) USED TO CONTROL THE NUMBER OF 8831
! RECORDS PRINTED ON EACH PAGE OF THE LOG FILE 8832
!===== 8833
!                                                         8834
!===== 8835
SET ABORT 8836
READY REFERENCES          SHARED WRITE 8837
READY REFERENCES FORM    SHARED READ 8838
MSG1 = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY" 8839
MSG2 = "THE FIRST LINE OF THE DOCUMENT TITLE SHOULD NOT END WITH " | 8840
      "A HYPHENATED WORD" 8841
MSG3 = "ORIGINATING ORGANIZATION IS NOT VALID -- NOT IN " | 8842
      "REFERENCE SOURCE TABLE" 8843
MSG4 = "DOCUMENT DATE IS NOT VALID -- MONTH MUST BE JAN, FEB, MAR, ETC." 8844
MSG5 = "DOCUMENT DATE IS NOT VALID --" 8845
MSG5A = "EXCEEDS THE NUMBER OF DAYS IN" 8846
MSG6 = "DOCUMENT DATE IS NOT VALID -- YEAR MUST BE GREATER THAN 1960" 8847
MSG7 = "THIS TITLE, ORGANIZATION AND DATE ALREADY EXIST IN DOMAIN " | 8848
      "REFERENCES" 8849
MSG8 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A" 8850
ICNT = 0 8851
JCNT = 0 8852
!                                                         8853
!===== 8854
! PRIMARY LOOP TO STORE REFERENCES 8855
!===== 8856
! THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS: 8857
! 1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR 8858
!    CORRECTION OF INVALID INFORMATION 8859
!===== 8860
!===== 8861
!===== 8862

```


Datatrieve Procedure REF_STORE (cont.)

```

!      2. IF TCONTINUE NE "A", SECTION TO REARRANGE THE VARIABLES      8863
!      ASSOCIATED WITH AUTHORS                                          8864
!      3. IF TCONTINUE NE "A", SECTION TO STORE RECORD IN DOMAIN      8865
!      REFERENCES AFTER VALIDATION TESTS HAVE BEEN PASSED             8866
!      4. IF TCONTINUE NE "A", SECTION TO PRINT DATA STORED IN DOMAIN 8867
!      REFERENCES FOR INCLUSION IN THE SESSION LOG FILE                8868
!      5. IF TCONTINUE = "A", SECTION TO PRINT MESSAGE THAT DATA CURRENTLY 8869
!      ON FORM HAS NOT BEEN STORED                                     8870
!      6. IF TCONTINUE = "A", SECTION TO REQUEST RESPONSE TO CONTINUE 8871
!      PROCEDURE OR EXIT TO MENU                                       8872
!                                                                      8873
=====                                                                8874
!                                                                      8875
TCONTINUE = "Y"                                                        8876
WHILE TCONTINUE = "Y"                                                  8877
BEGIN                                                                    8878
!                                                                      8879
!                                                                      8880
!                                                                      8881
=====                                                                8882
!      LOOP TO DISPLAY A BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE 8883
!      FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA 8884
!                                                                      8885
=====                                                                8886
!                                                                      8887
FLG1 = "N"                                                              8888
IMSG = 1                                                                8889
WHILE FLG1 = "N"                                                        8890
BEGIN                                                                    8891
    IF IMSG = 1 THEN TMSG = TMSG1                                       8892
    IF IMSG = 2 THEN TMSG = TMSG2                                       8893
    IF IMSG = 3 THEN TMSG = TMSG3                                       8894
    IF IMSG = 4 THEN TMSG = TMSG4                                       8895
    IF IMSG = 5 THEN TMSG = TMSG5 || " " | TDAY | " " | TMSG5A ||      8896
    " " | TMONTH VIA MONTH_TABLE                                         8897
    IF IMSG = 6 THEN TMSG = TMSG6                                       8898
    IF IMSG = 7 THEN TMSG = TMSG7                                       8899
    IF IMSG = 8 THEN TMSG = TMSG8                                       9000
!                                                                      9001
!                                                                      9002
=====                                                                9003
!      THIS SECTION DISPLAYS THE STORE REFERENCES FORM AND RETRIEVES      9004
!      THE DATA ENTERED ON THE FORM                                     9005
!                                                                      9006
=====                                                                9007
!                                                                      9008
!                                                                      9009
FOR FIRST 1 REFERENCES_FORM                                            9010
BEGIN                                                                    9011
    DISPLAY_FORM REFERENCES STO FORM IN                                  9012
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING                        9013

```

Datatrieve Procedure REF_STORE (cont.)

!!!

IF I = 5 THEN TEMP = TTITLE1	8965
IF I = 6 THEN TEMP = TTITLE2	8966
IF I = 7 THEN TEMP = TSOURCE	8967
IF I = 8 THEN TEMP = TDOCNO	8968
IF I = 9 THEN TEMP = TCONTNO	8969
IF TEMP NE " " THEN	8970
BEGIN	8971
J = 1	8972
WHILE FN\$STR_EXTRACT(TEMP, J, 1) = " "	8973
BEGIN	8974
J = J + 1	8975
END	8976
TEMP = FN\$STR_EXTRACT(TEMP, J, 112 - J + 1)	8977
END	8978
IF I = 1 THEN TAUTOR1 = TEMP	8979
IF I = 2 THEN TAUTOR2 = TEMP	8980
IF I = 3 THEN TAUTOR3 = TEMP	8981
IF I = 4 THEN TAUTOR4 = TEMP	8982
IF I = 5 THEN TTITLE1 = TEMP	8983
IF I = 6 THEN TTITLE2 = TEMP	8984
IF I = 7 THEN TSOURCE = TEMP	8985
IF I = 8 THEN TDOCNO = TEMP	8986
IF I = 9 THEN TCONTNO = TEMP	8987
I = I + 1	8988
END	8989
END	8990
	8991
	8992
=====	8993
IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO	8994
VERIFY THE FOLLOWING:	8995
1. TTITLE1 DOES NOT END WITH A HYPENATED WORD	8996
2. TSOURCE IS IN REFERENCE_SOURCE_TABLE	8997
3. TMONTH IS IN MONTH_TABLE	8998
4. NDAY IS LESS THAN OR EQUAL TO THE NUMBER OF DAYS IN	8999
TMONTH	9000
5. NYEAR IS GREATER THAN OR EQUAL TO 1960	9001
6. TTITLE, TSOURCE AND TDOCDATE DO NOT ALREADY EXIST IN	9002
DOMAIN REFERENCES	9003
7. TCONTINUE IS "Y", "N" OR "A"	9004
	9005
	9006
=====	9007
FLG1 = "Y"	9008
IF TCONTINUE NE "A" THEN	9009
BEGIN	9010
J = 80	9011
WHILE FN\$STR_EXTRACT(TTITLE1, J, 1) = " "	9012
BEGIN	9013
J = J - 1	9014
	9015

Datatrieve Procedure REF_STORE (cont.)

END	9016
IF FN\$STR_EXTRACT(TTITLE1, J, 1) = "-" AND	9017
FN\$STR_EXTRACT(TTITLE1, J - 1, 1) NE " " AND	9018
FN\$STR_EXTRACT(TTITLE1, J - 2, 2) NE " -" THEN	9019
BEGIN	9020
FLG1 = "N"	9021
IMSG = 2	9022
END	9023
TTITLE = TTITLE1 " " TTITLE2	9024
IF FLG1 = "Y" THEN	9025
BEGIN	9026
IF TSOURCE NOT IN REFERENCE_SOURCE_TABLE THEN	9027
BEGIN	9028
FLG1 = "N"	9029
IMSG = 3	9030
END	9031
END	9032
IF FLG1 = "Y" THEN	9033
BEGIN	9034
TMONTH = FN\$STR_EXTRACT(TDOCDATE, 3, 3)	9035
IF TMONTH NOT IN MONTH_TABLE THEN	9036
BEGIN	9037
FLG1 = "N"	9038
IMSG = 4	9039
END	9040
END	9041
IF FLG1 = "Y" THEN	9042
BEGIN	9043
TDAY = FN\$STR_EXTRACT(TDOCDATE, 1, 2)	9044
NDAY = TDAY	9045
IF NDAY LT 10 THEN TDAY = "0" TDAY	9046
IF (TMONTH = "JAN" AND NDAY GT 31) OR	9047
(TMONTH = "FEB" AND NDAY GT 29) OR	9048
(TMONTH = "MAR" AND NDAY GT 31) OR	9049
(TMONTH = "APR" AND NDAY GT 30) OR	9050
(TMONTH = "MAY" AND NDAY GT 31) OR	9051
(TMONTH = "JUN" AND NDAY GT 30) OR	9052
(TMONTH = "JUL" AND NDAY GT 31) OR	9053
(TMONTH = "AUG" AND NDAY GT 31) OR	9054
(TMONTH = "SEP" AND NDAY GT 30) OR	9055
(TMONTH = "OCT" AND NDAY GT 31) OR	9056
(TMONTH = "NOV" AND NDAY GT 30) OR	9057
(TMONTH = "DEC" AND NDAY GT 31) THEN	9058
BEGIN	9059
FLG1 = "N"	9060
IMSG = 5	9061
END	9062
END	9063
IF FLG1 = "Y" THEN	9064
BEGIN	9065
TYEAR = FN\$STR_EXTRACT(TDOCDATE, 6, 4)	9066

```

NYEAR = TYEAR
IF NYEAR LT 1960 THEN
  BEGIN
    FLG1 = "N"
    IMSG = 6
  END
END
IF FLG1 = "Y" THEN
  BEGIN
    FOR REFERENCES WITH DOCUMENT_TITLE = TTITLE      AND
                        DOCUMENT_SOURCE = TSOURCE      AND
                        DOCUMENT_DATE   = TDAY          AND
                        TMONTH          = "-"          AND
                        TYEAR           = "-"          AND
  BEGIN
    FLG1 = "N"
    IMSG = 7
  END
END
IF FLG1 = "Y" THEN
  BEGIN
    IF TCONTINUE NE "Y" AND
       TCONTINUE NE "N" AND
       TCONTINUE NE "A" THEN
      BEGIN
        FLG1 = "N"
        IMSG = 8
      END
    END
  END
END
END
=====
IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE
VARIABLES TAUTHOR1, TAUTHOR2, TAUTHOR3 AND TAUTHOR4 SO THAT THE
FIRST VALUE WHICH IS NOT BLANK IS TAUTHOR1, THE SECOND VALUE WHICH
IS NOT BLANK IS TAUTHOR2, ETC.
=====
IF TCONTINUE NE "A" THEN
  BEGIN
    TEMP = TAUTHOR1 || "$#" || TAUTHOR2 || "$#" ||
           TAUTHOR3 || "$#" || TAUTHOR4 || "$#"
    I = 1
    WHILE I LE 4
      BEGIN
        J = 1
        WHILE J = 1

```

Datatrieve Procedure REF_STORE (cont.)

```

BEGIN
    J = FN$STR_LOC(TEMP, "$$")
    IF J = 0 THEN TTAUTHOR = " "
    IF J = 1 THEN TEMP = FN$STR_EXTRACT(TEMP, 4, 109)
    IF J GT 1 THEN
        BEGIN
            TTAUTHOR = FN$STR_EXTRACT(TEMP, 1, J - 1)
            TEMP = FN$STR_EXTRACT(TEMP, J, 112 - J + 1)
        END
    END
    IF I = 1 THEN TAUTHOR1 = TTAUTHOR
    IF I = 2 THEN TAUTHOR2 = TTAUTHOR
    IF I = 3 THEN TAUTHOR3 = TTAUTHOR
    IF I = 4 THEN TAUTHOR4 = TTAUTHOR
    I = I + 1
END
END

```

=====

IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN
DOMAIN REFERENCES

=====

```

IF TCONTINUE NE "A" THEN
    BEGIN
        TSEQNO = 1
        FOR REFERENCES WITH DOCUMENT_SOURCE = TSOURCE SORTED BY
            REFERENCE_NUMBER
        BEGIN
            TSEQNO = SEQNO + 1
        END
        CAL = "NOW"
        STORE REFERENCES USING
        BEGIN
            DATE_CREATED = CAL
            SOURCE_ABBREVIATION = TSOURCE VIA REFERENCE_ABBREV_TABLE
            SEQUENCE_NUMBER = TSEQNO
            AUTHOR1 = TAUTHOR1
            AUTHOR2 = TAUTHOR2
            AUTHOR3 = TAUTHOR3
            AUTHOR4 = TAUTHOR4
            DOCUMENT_TITLE = TTITLE
            DOCUMENT_SOURCE = TSOURCE
            DOCUMENT_NUMBER = TDOCNO
            DOCUMENT_DATE = TDAY | "-" | TMONTH | "-" | TYEAR
            CONTRACT_NUMBER = TCONTNO
        END
    END
END

```

Datatrieve Procedure REF_STORE (cont.)

```

=====
IF TCONTINUE IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN
REFERENCES IS PRINTED. THE OUTPUT OF THE PRINT STATEMENTS WILL BE
INCLUDED IN THE SESSION LOG FILE WHICH IS OPENED BY THE CALLING
COMMAND PROCEDURE.
=====

```

```

IF TCONTINUE NE "A" THEN

```

```

  BEGIN

```

```

    ICNT = ICNT + 1

```

```

    JCNT = JCNT + 1

```

```

    IF JCNT = 1 THEN PRINT NEW PAGE

```

```

    FOR REFERENCES WITH DOCUMENT_TITLE = TTITLE      AND
                        DOCUMENT_SOURCE = TSOURCE     AND
                        DOCUMENT_DATE   = TDAY        | "-" |
                        TMONTH          | "-" |
                        TYEAR

```

```

    PRINT SKIP 4,

```

```

      COL 1, "RECORD NO.",          SPACE 1,
            ICNT (-) USING ZZ9, SKIP 1,

```

```

      COL 1, "=====", SPACE 0,
            "=====", SPACE 0,
            "=====", SPACE 0,
            "=====", SKIP 2,

```

```

      COL 3, "DATE CREATED      :",          SPACE 1,
            DATE_CREATED (-) USING X(23), SKIP 1,

```

```

      COL 3, "REFERENCE NUMBER :",          SPACE 1,
            REFERENCE_NUMBER (-) USING X(5), SKIP 1,

```

```

      COL 3, "AUTHOR1          :",          SPACE 1,
            AUTHOR1 (-) USING T(25), SKIP 1,

```

```

      COL 3, "AUTHOR2          :",          SPACE 1,
            AUTHOR2 (-) USING T(25), SKIP 1,

```

```

      COL 3, "AUTHOR3          :",          SPACE 1,
            AUTHOR3 (-) USING T(25), SKIP 1,

```

```

      COL 3, "AUTHOR4          :",          SPACE 1,
            AUTHOR4 (-) USING T(25), SKIP 1,

```

```

      COL 3, "DOCUMENT TITLE   :",          SPACE 1,
            DOCUMENT_TITLE (-) USING T(55), SKIP 1,

```

```

      COL 3, "DOCUMENT SOURCE  :",          SPACE 1,
            DOCUMENT_SOURCE (-) USING X(30), SKIP 1,

```

```

      COL 3, "DOCUMENT NUMBER  :",          SPACE 1,
            DOCUMENT_NUMBER (-) USING X(30), SKIP 1,

```

```

      COL 3, "DOCUMENT DATE    :",          SPACE 1,
            DOCUMENT_DATE (-) USING X(20), SKIP 1,

```

```

      COL 3, "CONTRACT NUMBER  :",          SPACE 1,
            CONTRACT_NUMBER (-) USING X(20), SKIP 2,

```

```

      COL 1, "=====", SPACE 0,

```

Datatrieve Procedure REF_STORE (cont.)

```

=====, SPACE 0, 9220
=====, SPACE 0, 9221
===== 9222
IF JCNT = 2 THEN JCNT = 0 9223
:BELL 9224
END 9225
! 9226
! 9227
! 9228
! 9229
! 9230
! 9231
! 9232
! 9233
! 9234
! 9235
! 9236
! 9237
! 9238
! 9239
! 9240
! 9241
! 9242
! 9243
! 9244
! 9245
! 9246
! 9247
! 9248
! 9249
! 9250
! 9251
! 9252
! 9253
! 9254
! 9255
! 9256
! 9257
! 9258
! 9259
! 9260
! 9261
! 9262
! 9263
! 9264
! 9265
! 9266
! 9267
! 9268
! 9269
! 9270

```

IF TCONTINUE IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED. THIS MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE.

```

=====
IF TCONTINUE = "A" THEN
  BEGIN
    JCNT = JCNT + 1
    IF JCNT = 1 THEN PRINT NEW_PAGE
    PRINT SKIP 4,
      COL 1, "=====, SPACE 0,
      "=====, SKIP 4,
      COL 9, "*****", SKIP 2,
      COL 9, "***** RECORD NOT STORED *****", SKIP 2,
      COL 9, "*****", SKIP 4,
      COL 1, "=====, SPACE 0,
      "=====
    IF JCNT = 2 THEN JCNT = 0
  END

```

IF TCONTINUE EQUALS "A", A RESPONSE IS REQUESTED TO EITHER CONTINUE THE PROCEDURE TO STORE REFERENCES OR EXIT TO THE MENU

```

=====
IF TCONTINUE = "A" THEN
  BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    TCONTINUE = "X"
    WHILE TCONTINUE NE "Y" AND
      TCONTINUE NE "N"
      BEGIN
        PRINT SKIP 2,
          "Do you wish to continue entering REFERENCES?",
          SKIP 1
        TCONTINUE = FN$UPCASE(*."Y or N")

```


Datatrieve Procedure REF_STORE (cont.)

PRINT " "	9271
END	9272
END	9273
END	9274
END-PROCEDURE	9275

Datatrieve Procedure S132

DEFINE PROCEDURE S132	9276
SET COLUMNS PAGE = 132	9277
FN\$WIDTH(132)	9278
END-PROCEDURE	9279

Datatrieve Procedure S80

DEFINE PROCEDURE S80	9280
SET COLUMNS PAGE = 80	9281
FN\$WIDTH(80)	9282
END-PROCEDURE	9283

Datatrieve Procedure SYS MODIFY

```

DEFINE PROCEDURE SYS_MODIFY
!
!
=====
!
!   VARIABLES ASSOCIATED WITH SYSTEM AND SYSTEM_NAME
!
=====
DECLARE TSYSTEM          PIC X(4).
DECLARE TSYSNAME        PIC X(80).
DECLARE TTSYSNAME       PIC X(80).
!
!
=====
!
!   VARIABLES ASSOCIATED WITH FMEA_ITEMS
!
=====
!
DECLARE TITEM1          PIC X(4).
DECLARE TITEM2          PIC X(4).
DECLARE TITEM3          PIC X(4).
DECLARE TITEM4          PIC X(4).
DECLARE TITEM5          PIC X(4).
DECLARE TITEM6          PIC X(4).
DECLARE TITEM7          PIC X(4).
DECLARE TITEM8          PIC X(4).
DECLARE TITEM9          PIC X(4).
DECLARE TITEM10         PIC X(4).
DECLARE TITEM11         PIC X(4).
DECLARE TITEM12         PIC X(4).
DECLARE TITEM13         PIC X(4).
DECLARE TITEM14         PIC X(4).
DECLARE TITEM15         PIC X(4).
DECLARE TTITEM1         PIC X(4).
DECLARE TTITEM2         PIC X(4).
DECLARE TTITEM3         PIC X(4).
DECLARE TTITEM4         PIC X(4).
DECLARE TTITEM5         PIC X(4).
DECLARE TTITEM6         PIC X(4).
DECLARE TTITEM7         PIC X(4).
DECLARE TTITEM8         PIC X(4).
DECLARE TTITEM9         PIC X(4).
DECLARE TTITEM10        PIC X(4).
DECLARE TTITEM11        PIC X(4).
DECLARE TTITEM12        PIC X(4).
DECLARE TTITEM13        PIC X(4).
DECLARE TTITEM14        PIC X(4).
DECLARE TTITEM15        PIC X(4).
DECLARE TITEM           PIC X(4).
!

```

Datatrieve Procedure SYS_MODIFY (cont.)

```

!
=====
!
!  VARIABLES ASSOCIATED WITH REFERENCES
!
=====
!
DECLARE TREF1      PIC X(5).
DECLARE TREF2      PIC X(5).
DECLARE TREF3      PIC X(5).
DECLARE TREF4      PIC X(5).
DECLARE TREF5      PIC X(5).
DECLARE TREF6      PIC X(5).
DECLARE TREF7      PIC X(5).
DECLARE TREF8      PIC X(5).
DECLARE TREF9      PIC X(5).
DECLARE TREF10     PIC X(5).
DECLARE TTREF1     PIC X(5).
DECLARE TTREF2     PIC X(5).
DECLARE TTREF3     PIC X(5).
DECLARE TTREF4     PIC X(5).
DECLARE TTREF5     PIC X(5).
DECLARE TTREF6     PIC X(5).
DECLARE TTREF7     PIC X(5).
DECLARE TTREF8     PIC X(5).
DECLARE TTREF9     PIC X(5).
DECLARE TTREF10    PIC X(5).
DECLARE TREF       PIC X(5).
!
!
=====
!
!  VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR FMEA_ITEMS AND
!  REFERENCES
!
=====
!
DECLARE TEMP      PIC X(60).
!
!
=====
!
!  VARIABLES USED AS FLAGS OR CONDITION INDICATORS
!
=====
!
DECLARE TCONTINUE1 PIC X(1).
DECLARE TCONTINUE2 PIC X(1).
DECLARE MSG        PIC 9(2).
DECLARE FLG1       PIC X(1).
DECLARE FLG2       PIC X(1).

```

9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385

Datatrieve Procedure SYS_MODIFY (cont.)

DECLARE FLG3	PIC X(1).	9386
DECLARE FLG4	PIC X(1).	9387
DECLARE FLG5	PIC X(1).	9388
!		9389
!		9390
=====		9391
!		9392
! VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM		9393
!		9394
=====		9395
!		9396
DECLARE TMSG	PIC X(80).	9397
DECLARE TMSG1	PIC X(80).	9398
DECLARE TMSG2	PIC X(80).	9399
DECLARE TMSG3	PIC X(80).	9400
DECLARE TMSG4	PIC X(80).	9401
DECLARE TMSG4A	PIC X(80).	9402
DECLARE TMSG5	PIC X(80).	9403
DECLARE TMSG5A	PIC X(80).	9404
DECLARE TMSG6	PIC X(80).	9405
DECLARE TMSG7	PIC X(80).	9406
DECLARE TMSG8	PIC X(80).	9407
DECLARE TMSG9	PIC X(80).	9408
DECLARE TMSG10	PIC X(80).	9409
!		9410
!		9411
=====		9412
!		9413
! VARIABLES USED AS COUNTERS		9414
!		9415
=====		9416
!		9417
DECLARE ICNT	PIC 9(4).	9418
DECLARE JCNT	PIC 9(4).	9419
DECLARE KCNT	PIC 9(4).	9420
DECLARE I	PIC 9(4).	9421
!		9422
!		9423
=====		9424
!		9425
! VARIABLES USED TO TEMPORARILY STORE SYSTEM DATA FOR COMPARISON OF		9426
! INITIAL AND MODIFIED VALUES		9427
!		9428
=====		9429
!		9430
DECLARE HDCREATED	USAGE DATE	9431
	EDIT STRING X(23).	9432
DECLARE HSYSTEM	PIC X(4).	9433
DECLARE HSYSNAME	PIC X(80).	9434
DECLARE HITEM1	PIC X(4).	9435
DECLARE HITEM2	PIC X(4).	9436

Datatrieve Procedure SYS_MODIFY (cont.)

DECLARE HITEM3	PIC X(4).	9437
DECLARE HITEM4	PIC X(4).	9438
DECLARE HITEM5	PIC X(4).	9439
DECLARE HITEM6	PIC X(4).	9440
DECLARE HITEM7	PIC X(4).	9441
DECLARE HITEM8	PIC X(4).	9442
DECLARE HITEM9	PIC X(4).	9443
DECLARE HITEM10	PIC X(4).	9444
DECLARE HITEM11	PIC X(4).	9445
DECLARE HITEM12	PIC X(4).	9446
DECLARE HITEM13	PIC X(4).	9447
DECLARE HITEM14	PIC X(4).	9448
DECLARE HITEM15	PIC X(4).	9449
DECLARE HREF1	PIC X(5).	9450
DECLARE HREF2	PIC X(5).	9451
DECLARE HREF3	PIC X(5).	9452
DECLARE HREF4	PIC X(5).	9453
DECLARE HREF5	PIC X(5).	9454
DECLARE HREF6	PIC X(5).	9455
DECLARE HREF7	PIC X(5).	9456
DECLARE HREF8	PIC X(5).	9457
DECLARE HREF9	PIC X(5).	9458
DECLARE HREF10	PIC X(5).	9459
DECLARE HFIPCREATED	PIC X(3).	9460
DECLARE HDLASTMOD	USAGE DATE	9461
	EDIT_STRING X(23).	9462
DECLARE HMODPROC	PIC X(20).	9463
!		9464
!		9465
!=====		9466
!		9467
! VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE		9468
!		9469
!=====		9470
!		9471
DECLARE CAL	USAGE DATE	9472
	EDIT_STRING X(23).	9473
!		9474
!		9475
!=====		9476
!		9477
! READY THE DOMAINS REFERENCES, SYSTEMS AND SYSTEMS_FORM		9478
!		9479
!=====		9480
!		9481
SET ABORT		9482
READY SYSTEMS	SHARED WRITE	9483
READY REFERENCES	SHARED READ	9484
READY SYSTEMS_FORM	SHARED READ	9485
!		9486
!		9487

Datatrieve Procedure SYS_MODIFY (cont.)

```

=====
!
!   INITIALIZE THE MESSAGE VARIABLES, INITIALIZE THE COUNTER (KCNT) USED
!   FOR NUMBERING THE LOG FILE RECORDS
!
=====
!
TMSG1  = "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"
TMSG2  = "DATA MUST BE ENTERED IN AT LEAST ONE OF THE FIELDS OR CONTINUE "
        "MUST BE N"
TMSG3  = "SYSTEM IS NOT VALID -- NOT IN DOMAIN SYSTEMS"
TMSG4  = "FMEA ITEM IS NOT VALID --"
TMSG4A = "NOT IN FMEA_ITEM_NAME_TABLE"
TMSG5  = "REFERENCE DOCUMENT IS NOT VALID --"
TMSG5A = "DOES NOT EXIST IN DOMAIN REFERENCES"
TMSG6  = "CONTINUE IS NOT VALID -- MUST BE Y OR N"
TMSG7  = "NO RECORDS HAVE BEEN FOUND WITH THE DATA INDICATED ABOVE"
TMSG8  = "ENTER MODIFICATIONS IN APPROPRIATE FIELDS AND PRESS RETURN KEY"
TMSG9  = "SYSTEM NAME IS NOT VALID -- ALREADY EXISTS IN DOMAIN SYSTEMS"
TMSG10 = "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"
KCNT   = 0

!
!
=====
!
!   PRIMARY LOOP TO MODIFY SYSTEMS
!
!   THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:
!       1. LOOP TO REQUEST SEARCH DATA, TEST VALUES, PROMPT FOR CORRECTION
!          OF INVALID INFORMATION AND COUNT THE RECORDS WHICH MATCH THE
!          SPECIFIED INPUT FIELDS
!       2. IF TCONTINUE1 NE "N", SECTION TO DISPLAY THE MATCHING RECORDS
!          ONE AT A TIME FOR POSSIBLE MODIFICATION (THIS SECTION IS
!          TERMINATED WHEN TCONTINUE2 = "N")
!       3. IF TCONTINUE1 = "N", SECTION TO REQUEST RESPONSE TO CONTINUE
!          PROCEDURE OR EXIT TO MENU
!
=====
!
TCONTINUE1 = "Y"
WHILE TCONTINUE1 = "Y"
BEGIN
!
!   =====
!   !
!   !   LOOP TO DISPLAY BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE
!   !   FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA
!   !
!   =====
!

```


! ! ! ! !

```

FLG1 = "N"
IMSG = 1
WHILE FLG1 = "N"
  BEGIN
    IF IMSG = 1 THEN TMSG = TMSG1
    IF IMSG = 2 THEN TMSG = TMSG2
    IF IMSG = 3 THEN TMSG = TMSG3
    IF IMSG = 4 THEN TMSG = TMSG4 || " " | TITEM | " " | TMSG4A
    IF IMSG = 5 THEN TMSG = TMSG5 || " " | TREF | " " | TMSG5A
    IF IMSG = 6 THEN TMSG = TMSG6
    IF IMSG = 7 THEN TMSG = TMSG7
    IF IMSG = 8 THEN TMSG = TMSG8
    IF IMSG = 9 THEN TMSG = TMSG9
    IF IMSG = 10 THEN TMSG = TMSG10

    =====
    |
    | THIS SECTION DISPLAYS THE FIND SYSTEMS FORM AND RETRIEVES
    | THE DATA ENTERED ON THE FORM
    |
    =====

  FOR FIRST 1 SYSTEMS_FORM
  BEGIN
    DISPLAY FORM SYSTEMS FIN FORM IN
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING
    BEGIN
      IF IMSG NE 1 THEN
        BEGIN
          PUT FORM SYSTEM = TSYSTEM
          PUT FORM SYSTEM_NAME = TSYNAME
          PUT FORM ITEM_1 = TITEM1
          PUT FORM ITEM_2 = TITEM2
          PUT FORM ITEM_3 = TITEM3
          PUT FORM ITEM_4 = TITEM4
          PUT FORM ITEM_5 = TITEM5
          PUT FORM ITEM_6 = TITEM6
          PUT FORM ITEM_7 = TITEM7
          PUT FORM ITEM_8 = TITEM8
          PUT FORM ITEM_9 = TITEM9
          PUT FORM ITEM_10 = TITEM10
          PUT FORM ITEM_11 = TITEM11
          PUT FORM ITEM_12 = TITEM12
          PUT FORM ITEM_13 = TITEM13
          PUT FORM ITEM_14 = TITEM14
          PUT FORM ITEM_15 = TITEM15
          PUT FORM REF_1 = TREF1
          PUT FORM REF_2 = TREF2
          PUT FORM REF_3 = TREF3
          PUT FORM REF_4 = TREF4

```

Datatrieve Procedure SYS_MODIFY (cont.)

```

          PUT_FORM REF_5      = TREF5      9590
          PUT_FORM REF_6      = TREF6      9591
          PUT_FORM REF_7      = TREF7      9592
          PUT_FORM REF_8      = TREF8      9593
          PUT_FORM REF_9      = TREF9      9594
          PUT_FORM REF_10     = TREF10     9595
      END                                     9596
      PUT_FORM CONTINUE = TCONTINUE1      9597
      PUT_FORM MESSAGE  = TMSG            9598
  END RETRIEVE USING                       9599
  BEGIN                                    9600
      TSYSTEM      = GET_FORM SYSTEM      9601
      TSYSNAME     = GET_FORM SYSTEM_NAME 9602
      TITEM1       = GET_FORM ITEM_1      9603
      TITEM2       = GET_FORM ITEM_2      9604
      TITEM3       = GET_FORM ITEM_3      9605
      TITEM4       = GET_FORM ITEM_4      9606
      TITEM5       = GET_FORM ITEM_5      9607
      TITEM6       = GET_FORM ITEM_6      9608
      TITEM7       = GET_FORM ITEM_7      9609
      TITEM8       = GET_FORM ITEM_8      9610
      TITEM9       = GET_FORM ITEM_9      9611
      TITEM10      = GET_FORM ITEM_10     9612
      TITEM11      = GET_FORM ITEM_11     9613
      TITEM12      = GET_FORM ITEM_12     9614
      TITEM13      = GET_FORM ITEM_13     9615
      TITEM14      = GET_FORM ITEM_14     9616
      TITEM15      = GET_FORM ITEM_15     9617
      TREF1        = GET_FORM REF_1       9618
      TREF2        = GET_FORM REF_2       9619
      TREF3        = GET_FORM REF_3       9620
      TREF4        = GET_FORM REF_4       9621
      TREF5        = GET_FORM REF_5       9622
      TREF6        = GET_FORM REF_6       9623
      TREF7        = GET_FORM REF_7       9624
      TREF8        = GET_FORM REF_8       9625
      TREF9        = GET_FORM REF_9       9626
      TREF10       = GET_FORM REF_10      9627
      TCONTINUE1   = GET_FORM CONTINUE    9628
  END                                     9629
  END                                     9630
  !                                     9631
  !                                     9632
  !                                     9633
  !                                     9634
  !                                     9635
  !                                     9636
  !                                     9637
  !                                     9638
  !                                     9639
  !                                     9640
  IF TCONTINUE1 NE "N" AND

```

IF TCONTINUE1 IS NOT EQUAL TO "N", ANY LEADING BLANKS WHICH WERE INADVERTENTLY ENTERED IN TSYNAME ARE REMOVED

Datatrieve Procedure SYS_MODIFY (cont.)

```

TSYSNAME  NE " " THEN
BEGIN
  I = 1
  WHILE FN$STR_EXTRACT(TSYSNAME, I, 1) = " "
  BEGIN
    I = I + 1
  END
  TSYSNAME = FN$STR_EXTRACT(TSYSNAME, I, 80 - I + 1)
END

```

```

=====
IF TCONTINUE1 IS NOT EQUAL TO "N", TESTS ARE PERFORMED TO
VERIFY ANY DATA ENTERED ON THE FORM. AS APPROPRIATE, THE
PROCEDURE CHECKS ANY OR ALL OF THE FOLLOWING:

```

1. DATA HAS BEEN ENTERED IN AT LEAST ONE OF THE FIELDS
2. TSYSTEM IS IN DOMAIN SYSTEMS
3. ANY OF THE VARIABLES TITEM1 TO TITEM15 WHICH ARE NOT BLANK ARE IN TABLE FMEA ITEM NAME TABLE
4. ANY OF THE VARIABLES TREF1 TO TREF10 WHICH ARE NOT BLANK ARE IN DOMAIN REFERENCES
5. TCONTINUE1 IS "Y" OR "N"
6. AT LEAST ONE RECORD EXISTS WITH THE DATA SPECIFIED

```

=====
FLG1 = "Y"
IF TCONTINUE1 NE "N" THEN
BEGIN
  IF TSYSTEM = " " AND TSYSNAME = " " AND
  TITEM1 = " " AND TITEM2 = " " AND TITEM3 = " " AND
  TITEM4 = " " AND TITEM5 = " " AND TITEM6 = " " AND
  TITEM7 = " " AND TITEM8 = " " AND TITEM9 = " " AND
  TITEM10 = " " AND TITEM11 = " " AND TITEM12 = " " AND
  TITEM13 = " " AND TITEM14 = " " AND TITEM15 = " " AND
  TREF1 = " " AND TREF2 = " " AND TREF3 = " " AND
  TREF4 = " " AND TREF5 = " " AND TREF6 = " " AND
  TREF7 = " " AND TREF8 = " " AND TREF9 = " " AND
  TREF10 = " " THEN
  BEGIN
    FLG1 = "N"
    MSG = 2
  END
  IF FLG1 = "Y" AND
  TSYSTEM NE " " THEN
  BEGIN
    FLG2 = "N"
    FOR SYSTEMS WITH SYSTEM = TSYSTEM
    FLG2 = "Y"
    IF FLG2 = "N" THEN

```

Datatrieve Procedure SYS_MODIFY (cont.)

```

        BEGIN
            FLG1 = "N"
            IMSG = 3
        END
    END
IF FLG1 = "Y" THEN
    BEGIN
        I = 1
        WHILE FLG1 = "Y" AND I LE 15
            BEGIN
                IF I = 1 THEN TITEM = TITEM1
                IF I = 2 THEN TITEM = TITEM2
                IF I = 3 THEN TITEM = TITEM3
                IF I = 4 THEN TITEM = TITEM4
                IF I = 5 THEN TITEM = TITEM5
                IF I = 6 THEN TITEM = TITEM6
                IF I = 7 THEN TITEM = TITEM7
                IF I = 8 THEN TITEM = TITEM8
                IF I = 9 THEN TITEM = TITEM9
                IF I = 10 THEN TITEM = TITEM10
                IF I = 11 THEN TITEM = TITEM11
                IF I = 12 THEN TITEM = TITEM12
                IF I = 13 THEN TITEM = TITEM13
                IF I = 14 THEN TITEM = TITEM14
                IF I = 15 THEN TITEM = TITEM15
                IF TITEM NE " " AND
                    TITEM NOT IN FMEA_ITEM_NAME_TABLE THEN
                    BEGIN
                        FLG1 = "N"
                        IMSG = 4
                    END
                I = I + 1
            END
        END
    END
IF FLG1 = "Y" THEN
    BEGIN
        I = 1
        WHILE FLG1 = "Y" AND I LE 10
            BEGIN
                IF I = 1 THEN TREF = TREF1
                IF I = 2 THEN TREF = TREF2
                IF I = 3 THEN TREF = TREF3
                IF I = 4 THEN TREF = TREF4
                IF I = 5 THEN TREF = TREF5
                IF I = 6 THEN TREF = TREF6
                IF I = 7 THEN TREF = TREF7
                IF I = 8 THEN TREF = TREF8
                IF I = 9 THEN TREF = TREF9
                IF I = 10 THEN TREF = TREF10
                IF TREF NE " " THEN
                    BEGIN

```

Datatrieve Procedure SYS_MODIFY (cont.)

FLG2 = "N"	9743
FOR REFERENCES WITH REFERENCE_NUMBER = TREF	9744
BEGIN	9745
FLG2 = "Y"	9746
END	9747
IF FLG2 = "N" THEN	9748
BEGIN	9749
FLG1 = "N"	9750
IMSG = 5	9751
END	9752
END	9753
I = I + 1	9754
END	9755
END	9756
IF FLG1 = "Y" THEN	9757
BEGIN	9758
IF TCONTINUE1 NE "Y" AND	9759
TCONTINUE1 NE "N" THEN	9760
BEGIN	9761
FLG1 = "N"	9762
IMSG = 6	9763
END	9764
END	9765
IF FLG1 = "Y" THEN	9766
BEGIN	9767
FLG2 = "N"	9768
IF TSYSTEM NE " " THEN	9769
BEGIN	9770
FLG2 = "Y"	9771
JCNT = 0	9772
FOR SYSTEMS WITH	9773
SYSTEM = TSYSTEM	9774
BEGIN	9775
FLG3 = "Y"	9776
:SYS_MODIFY_1	9777
IF FLG3 = "Y" THEN JCNT = JCNT + 1	9778
END	9779
END	9780
IF FLG2 = "N" THEN	9781
BEGIN	9782
JCNT = 0	9783
FOR SYSTEMS	9784
BEGIN	9785
FLG3 = "Y"	9786
:SYS_MODIFY_1	9787
IF FLG3 = "Y" THEN JCNT = JCNT + 1	9788
END	9789
END	9790
IF JCNT = 0 THEN	9791
BEGIN	9792
FLG1 = "N"	9793

Datatrieve Procedure SYS_MODIFY (cont.)

```

                                ICNT = ICNT + 1          9845
                                :SYS_MODIFY_2          9846
                                END                    9847
                                END                    9848
                                END                    9849
                                END                    9850
                                END                    9851
                                END                    9852
                                END                    9853
                                END                    9854
                                END                    9855
                                END                    9856
                                END                    9857
                                END                    9858
                                END                    9859
                                END                    9860
                                END                    9861
                                END                    9862
                                END                    9863
                                END                    9864
                                END                    9865
                                END                    9866
                                END                    9867
                                END                    9868
                                END                    9869
                                END                    9870
                                END                    9871
                                END                    9872
                                END                    9873
                                END                    9874
                                END                    9875
                                END                    9876
                                END                    9877

```

=====

IF TCONTINUE1 IS EQUAL TO "N", A RESPONSE IS REQUESTED TO EITHER
CONTINUE THE PROCEDURE TO MODIFY SYSTEMS OR EXIT TO THE MENU

=====

```

IF TCONTINUE1 = "N" THEN
  BEGIN
    PRINT NEW_PAGE
    :CLRSCRN
    TCONTINUE1 = "X"
    WHILE TCONTINUE1 NE "Y" AND
      TCONTINUE1 NE "N"
      BEGIN
        PRINT SKIP 2,
          "Do you wish to continue modifying", SKIP 1,
          "SYSTEMS?", SKIP 1
        TCONTINUE1 = FN$UPCASE(*."Y or N")
        PRINT " "
      END
    END
  END
END-PROCEDURE

```

Datatrieve Procedure SYS_MODIFY_1

```

DEFINE PROCEDURE SYS_MODIFY_1
!
!
!=====
!
! THIS SECTION DETERMINES IF A GIVEN RECORD MATCHES ALL OF THE FIELDS
! ENTERED ON THE FIND SYSTEMS FORM.  THE VALUES OF THE KEY FIELD
! (SYSTEM) IS NOT CHECKED IN THIS SECTION SINCE THIS VALUE HAS ALREADY
! BEEN MATCHED IN THE "FOR ..." STATEMENT WHICH ESTABLISHES THE RECORD
! STREAM.
!=====
!
IF TSYSNAME NE " " AND
SYSTEM_NAME NE TSYSNAME THEN FLG3 = "N"
IF TITEM1 NE " " AND
ITEM1 NE TITEM1 THEN FLG3 = "N"
IF TITEM2 NE " " AND
ITEM2 NE TITEM2 THEN FLG3 = "N"
IF TITEM3 NE " " AND
ITEM3 NE TITEM3 THEN FLG3 = "N"
IF TITEM4 NE " " AND
ITEM4 NE TITEM4 THEN FLG3 = "N"
IF TITEM5 NE " " AND
ITEM5 NE TITEM5 THEN FLG3 = "N"
IF TITEM6 NE " " AND
ITEM6 NE TITEM6 THEN FLG3 = "N"
IF TITEM7 NE " " AND
ITEM7 NE TITEM7 THEN FLG3 = "N"
IF TITEM8 NE " " AND
ITEM8 NE TITEM8 THEN FLG3 = "N"
IF TITEM9 NE " " AND
ITEM9 NE TITEM9 THEN FLG3 = "N"
IF TITEM10 NE " " AND
ITEM10 NE TITEM10 THEN FLG3 = "N"
IF TITEM11 NE " " AND
ITEM11 NE TITEM11 THEN FLG3 = "N"
IF TITEM12 NE " " AND
ITEM12 NE TITEM12 THEN FLG3 = "N"
IF TITEM13 NE " " AND
ITEM13 NE TITEM13 THEN FLG3 = "N"
IF TITEM14 NE " " AND
ITEM14 NE TITEM14 THEN FLG3 = "N"
IF TITEM15 NE " " AND
ITEM15 NE TITEM15 THEN FLG3 = "N"
IF TREF1 NE " " AND
REFERENCE1 NE TREF1 THEN FLG3 = "N"
IF TREF2 NE " " AND
REFERENCE2 NE TREF2 THEN FLG3 = "N"
IF TREF3 NE " " AND
REFERENCE3 NE TREF3 THEN FLG3 = "N"

```


Datatrieve Procedure SYS_MODIFY_1 (cont.)

IF TREF4	NE " "	AND	9929
REFERENCE4	NE TREF4	THEN FLG3 = "N"	9930
IF TREF5	NE " "	AND	9931
REFERENCE5	NE TREF5	THEN FLG3 = "N"	9932
IF TREF6	NE " "	AND	9933
REFERENCE6	NE TREF6	THEN FLG3 = "N"	9934
IF TREF7	NE " "	AND	9935
REFERENCE7	NE TREF7	THEN FLG3 = "N"	9936
IF TREF8	NE " "	AND	9937
REFERENCE8	NE TREF8	THEN FLG3 = "N"	9938
IF TREF9	NE " "	AND	9939
REFERENCE9	NE TREF9	THEN FLG3 = "N"	9940
IF TREF10	NE " "	AND	9941
REFERENCE10	NE TREF10	THEN FLG3 = "N"	9942
END-PROCEDURE			9943

Datatrieve Procedure SYS_MODIFY_2

```

DEFINE PROCEDURE SYS_MODIFY_2
!
!
=====
! THE FIELDS OF THE INCOMING SYSTEM RECORD ARE ASSIGNED TO VARIABLES FOR
! DISPLAY AND MODIFICATION.  THE FIELDS SYSTEM AND
! PROPAGATIONS FILE_CREATED ARE NOT ASSIGNED TO VARIABLES SINCE THESE
! FIELDS CANNOT BE MODIFIED.
!
=====
!
TTSYSNAME = SYSTEM_NAME
TTITEM1   = ITEM1
TTITEM2   = ITEM2
TTITEM3   = ITEM3
TTITEM4   = ITEM4
TTITEM5   = ITEM5
TTITEM6   = ITEM6
TTITEM7   = ITEM7
TTITEM8   = ITEM8
TTITEM9   = ITEM9
TTITEM10  = ITEM10
TTITEM11  = ITEM11
TTITEM12  = ITEM12
TTITEM13  = ITEM13
TTITEM14  = ITEM14
TTITEM15  = ITEM15
TTREF1    = REFERENCE1
TTREF2    = REFERENCE2
TTREF3    = REFERENCE3
TTREF4    = REFERENCE4
TTREF5    = REFERENCE5
TTREF6    = REFERENCE6
TTREF7    = REFERENCE7
TTREF8    = REFERENCE8
TTREF9    = REFERENCE9
TTREF10   = REFERENCE10
!
!
=====
! LOOP TO DISPLAY A SYSTEM RECORD USING A TDMS FORM, RETRIEVE DATA FROM
! THE FORM, TEST THE INCOMING INFORMATION AND REQUEST CORRECTION OF
! INVALID DATA
!
=====
!
FLG4 = "N"
IMSG = 8
WHILE FLG4 = "N"

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

BEGIN
    IF IMSG = 4 THEN TMSG = TMSG4 || " " | TITEM | " " | TMSG4A
    IF IMSG = 5 THEN TMSG = TMSG5 || " " | TREF | " " | TMSG5A
    IF IMSG = 8 THEN TMSG = TMSG8
    IF IMSG = 9 THEN TMSG = TMSG9
    IF IMSG = 10 THEN TMSG = TMSG10

=====
    THIS SECTION DISPLAYS THE MODIFY SYSTEMS FORM AND RETRIEVES THE
    DATA ENTERED ON THE FORM
=====

DISPLAY FORM SYSTEMS_MOD FORM IN
DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING
BEGIN
    PUT_FORM RECORD_NUMBER = ICNT
    PUT_FORM TOTAL_RECORDS = JCNT
    PUT_FORM SYSTEM = SYSTEM
    PUT_FORM SYSTEM_NAME = TTSYSNAME
    PUT_FORM ITEM_1 = TTITEM1
    PUT_FORM ITEM_2 = TTITEM2
    PUT_FORM ITEM_3 = TTITEM3
    PUT_FORM ITEM_4 = TTITEM4
    PUT_FORM ITEM_5 = TTITEM5
    PUT_FORM ITEM_6 = TTITEM6
    PUT_FORM ITEM_7 = TTITEM7
    PUT_FORM ITEM_8 = TTITEM8
    PUT_FORM ITEM_9 = TTITEM9
    PUT_FORM ITEM_10 = TTITEM10
    PUT_FORM ITEM_11 = TTITEM11
    PUT_FORM ITEM_12 = TTITEM12
    PUT_FORM ITEM_13 = TTITEM13
    PUT_FORM ITEM_14 = TTITEM14
    PUT_FORM ITEM_15 = TTITEM15
    PUT_FORM REF_1 = TTREF1
    PUT_FORM REF_2 = TTREF2
    PUT_FORM REF_3 = TTREF3
    PUT_FORM REF_4 = TTREF4
    PUT_FORM REF_5 = TTREF5
    PUT_FORM REF_6 = TTREF6
    PUT_FORM REF_7 = TTREF7
    PUT_FORM REF_8 = TTREF8
    PUT_FORM REF_9 = TTREF9
    PUT_FORM REF_10 = TTREF10
    IF IMSG = 8 THEN
        PUT_FORM CONTINUE = "Y"
    IF IMSG NE 8 THEN
        PUT_FORM CONTINUE = TCONTINUE2

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

      PUT FORM MESSAGE          = TMSG                      10046
END RETRIEVE USING              10047
BEGIN                           10048
  TTSYSNAME = GET_FORM SYSTEM_NAME 10049
  TTITEM1   = GET_FORM ITEM_1      10050
  TTITEM2   = GET_FORM ITEM_2      10051
  TTITEM3   = GET_FORM ITEM_3      10052
  TTITEM4   = GET_FORM ITEM_4      10053
  TTITEM5   = GET_FORM ITEM_5      10054
  TTITEM6   = GET_FORM ITEM_6      10055
  TTITEM7   = GET_FORM ITEM_7      10056
  TTITEM8   = GET_FORM ITEM_8      10057
  TTITEM9   = GET_FORM ITEM_9      10058
  TTITEM10  = GET_FORM ITEM_10     10059
  TTITEM11  = GET_FORM ITEM_11     10060
  TTITEM12  = GET_FORM ITEM_12     10061
  TTITEM13  = GET_FORM ITEM_13     10062
  TTITEM14  = GET_FORM ITEM_14     10063
  TTITEM15  = GET_FORM ITEM_15     10064
  TTREF1    = GET_FORM REF_1       10065
  TTREF2    = GET_FORM REF_2       10066
  TTREF3    = GET_FORM REF_3       10067
  TTREF4    = GET_FORM REF_4       10068
  TTREF5    = GET_FORM REF_5       10069
  TTREF6    = GET_FORM REF_6       10070
  TTREF7    = GET_FORM REF_7       10071
  TTREF8    = GET_FORM REF_8       10072
  TTREF9    = GET_FORM REF_9       10073
  TTREF10   = GET_FORM REF_10      10074
  TCONTINUE2 = GET_FORM CONTINUE  10075
END                               10076
!                               10077
!                               10078
===== 10079
|  IF TCONTINUE2 IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH WERE  10080
|  INADVERTENTLY ENTERED IN TTYSNAME ARE REMOVED                     10081
|  10082
|  10083
===== 10084
!                               10085
!                               10086
IF TCONTINUE2 NE "A" THEN      10087
  BEGIN                          10088
    I = 1                        10088
    WHILE FN$STR_EXTRACT(TTSYSNAME, I, 1) = " " 10089
      BEGIN                      10090
        I = I + 1                10091
      END                        10092
    TTYSNAME = FN$STR_EXTRACT(TTSYSNAME, I, 80 - I + 1) 10093
  END                            10094
!                               10095
!                               10096

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

=====
IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
VERIFY THE FOLLOWING:
1. TTSYSNAME DOES NOT ALREADY EXIST IN DOMAIN SYSTEMS
2. ANY OF THE VARIABLES TITEM1 TO TITEM15 WHICH ARE NOT
   BLANK ARE IN FMEA_ITEM_NAME_TABLE
3. ANY OF THE VARIABLES TREF1 TO TREF10 WHICH ARE NOT BLANK
   ARE IN DOMAIN REFERENCES
5. TCONTINUE2 IS "Y", "N" OR "A"
=====
10097
10098
10099
10100
10101
10102
10103
10104
10105
10106
10107
10108
10109
10110
10111
10112
10113
10114
10115
10116
10117
10118
10119
10120
10121
10122
10123
10124
10125
10126
10127
10128
10129
10130
10131
10132
10133
10134
10135
10136
10137
10138
10139
10140
10141
10142
10143
10144
10145
10146
10147

FLG4 = "Y"
IF TCONTINUE2 NE "A" THEN
  BEGIN
    IF TTSYSNAME NE SYSTEM_NAME THEN
      BEGIN
        HSYSTEM = SYSTEM
        FOR SYSTEMS WITH SYSTEM NE HSYSTEM
          IF SYSTEM_NAME = TTSYSNAME THEN
            BEGIN
              FLG4 = "N"
              IMSG = 9
            END
          END
        IF FLG4 = "Y" THEN
          BEGIN
            I = 1
            WHILE FLG4 = "Y" AND I LE 15
              BEGIN
                IF I = 1 THEN TITEM = TTITEM1
                IF I = 2 THEN TITEM = TTITEM2
                IF I = 3 THEN TITEM = TTITEM3
                IF I = 4 THEN TITEM = TTITEM4
                IF I = 5 THEN TITEM = TTITEM5
                IF I = 6 THEN TITEM = TTITEM6
                IF I = 7 THEN TITEM = TTITEM7
                IF I = 8 THEN TITEM = TTITEM8
                IF I = 9 THEN TITEM = TTITEM9
                IF I = 10 THEN TITEM = TTITEM10
                IF I = 11 THEN TITEM = TTITEM11
                IF I = 12 THEN TITEM = TTITEM12
                IF I = 13 THEN TITEM = TTITEM13
                IF I = 14 THEN TITEM = TTITEM14
                IF I = 15 THEN TITEM = TTITEM15
                IF TITEM NE " " AND
                   TITEM NOT IN FMEA_ITEM_NAME_TABLE THEN
                  BEGIN
                    FLG4 = "N"
                    IMSG = 4

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

                                END                                10148
                                I = I + 1                          10149
                                END                                10150
                                END                                10151
                                IF FLG4 = "Y" THEN                10152
                                BEGIN                                10153
                                I = 1                                10154
                                WHILE FLG4 = "Y" AND I LE 10      10155
                                BEGIN                                10156
                                IF I = 1 THEN TREF = TTREF1        10157
                                IF I = 2 THEN TREF = TTREF2        10158
                                IF I = 3 THEN TREF = TTREF3        10159
                                IF I = 4 THEN TREF = TTREF4        10160
                                IF I = 5 THEN TREF = TTREF5        10161
                                IF I = 6 THEN TREF = TTREF6        10162
                                IF I = 7 THEN TREF = TTREF7        10163
                                IF I = 8 THEN TREF = TTREF8        10164
                                IF I = 9 THEN TREF = TTREF9        10165
                                IF I = 10 THEN TREF = TTREF10       10166
                                IF TREF NE " " THEN                10167
                                BEGIN                                10168
                                FLG5 = "N"                          10169
                                FOR REFERENCES WITH REFERENCE_NUMBER = TREF 10170
                                BEGIN                                10171
                                FLG5 = "Y"                          10172
                                END                                10173
                                IF FLG5 = "N" THEN                10174
                                BEGIN                                10175
                                FLG4 = "N"                          10176
                                MSG = 5                             10177
                                END                                10178
                                END                                10179
                                I = I + 1                          10180
                                END                                10181
                                END                                10182
                                IF FLG4 = "Y" THEN                10183
                                BEGIN                                10184
                                IF TCONTINUE2 NE "Y" AND          10185
                                TCONTINUE2 NE "N" AND              10186
                                TCONTINUE2 NE "A" THEN            10187
                                BEGIN                                10188
                                FLG4 = "N"                          10189
                                MSG = 10                           10190
                                END                                10191
                                END                                10192
                                END                                10193
                                END                                10194
                                !                                  10195
                                !                                  10196
                                !=====                          10197
                                !                                  10198

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

! IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE      10199
! VARIABLES TTITEM1 TO TTITEM15 SO THAT THE FIRST VALUE WHICH IS NOT  10200
! BLANK IS TTITEM1, THE SECOND VALUE WHICH IS NOT BLANK IS TTITEM2, ETC. 10201
!                                                                           10202
! =====                                                             10203
!                                                                           10204
IF TCONTINUE2 NE "A" THEN                                           10205
  BEGIN                                                             10206
    TEMP = TTITEM1 || TTITEM2 || TTITEM3 || TTITEM4 || TTITEM5 || 10207
           TTITEM6 || TTITEM7 || TTITEM8 || TTITEM9 || TTITEM10 || 10208
           TTITEM11 || TTITEM12 || TTITEM13 || TTITEM14 || TTITEM15 10209
    TTITEM1 = FN$STR_EXTRACT(TEMP, 1, 4)                            10210
    TTITEM2 = FN$STR_EXTRACT(TEMP, 5, 4)                            10211
    TTITEM3 = FN$STR_EXTRACT(TEMP, 9, 4)                            10212
    TTITEM4 = FN$STR_EXTRACT(TEMP, 13, 4)                           10213
    TTITEM5 = FN$STR_EXTRACT(TEMP, 17, 4)                           10214
    TTITEM6 = FN$STR_EXTRACT(TEMP, 21, 4)                           10215
    TTITEM7 = FN$STR_EXTRACT(TEMP, 25, 4)                           10216
    TTITEM8 = FN$STR_EXTRACT(TEMP, 29, 4)                           10217
    TTITEM9 = FN$STR_EXTRACT(TEMP, 33, 4)                           10218
    TTITEM10 = FN$STR_EXTRACT(TEMP, 37, 4)                           10219
    TTITEM11 = FN$STR_EXTRACT(TEMP, 41, 4)                           10220
    TTITEM12 = FN$STR_EXTRACT(TEMP, 45, 4)                           10221
    TTITEM13 = FN$STR_EXTRACT(TEMP, 49, 4)                           10222
    TTITEM14 = FN$STR_EXTRACT(TEMP, 53, 4)                           10223
    TTITEM15 = FN$STR_EXTRACT(TEMP, 57, 4)                           10224
  END                                                                10225
!                                                                           10226
! =====                                                             10227
!                                                                           10228
! IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE      10229
! VARIABLES TTREF1 TO TTREF10 SO THAT THE FIRST VALUE WHICH IS NOT    10230
! BLANK IS TTREF1, THE SECOND VALUE WHICH IS NOT BLANK IS TTREF2, ETC. 10231
! ETC.                                                                10232
!                                                                           10233
! =====                                                             10234
!                                                                           10235
IF TCONTINUE2 NE "A" THEN                                           10236
  BEGIN                                                             10237
    TEMP = TTREF1 || TTREF2 || TTREF3 || TTREF4 || TTREF5 || 10238
           TTREF6 || TTREF7 || TTREF8 || TTREF9 || TTREF10 || 10239
    TTREF1 = FN$STR_EXTRACT(TEMP, 1, 5)                            10241
    TTREF2 = FN$STR_EXTRACT(TEMP, 6, 5)                            10242
    TTREF3 = FN$STR_EXTRACT(TEMP, 11, 5)                           10243
    TTREF4 = FN$STR_EXTRACT(TEMP, 16, 5)                           10244
    TTREF5 = FN$STR_EXTRACT(TEMP, 21, 5)                           10245
    TTREF6 = FN$STR_EXTRACT(TEMP, 26, 5)                           10246
    TTREF7 = FN$STR_EXTRACT(TEMP, 31, 5)                           10247
    TTREF8 = FN$STR_EXTRACT(TEMP, 36, 5)                           10248
    TTREF9 = FN$STR_EXTRACT(TEMP, 41, 5)                           10249

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

      TTREF10 = FN$STR_EXTRACT(TEMP, 46, 5)                                10250
    END                                                                    10251
!                                                                           10252
!                                                                           10253
!=====                                                                  10254
!                                                                           10255
! IF TCONTINUE2 IS NOT EQUAL TO "A" AND NEW DATA HAS BEEN ENTERED IN AT 10256
! LEAST ONE OF THE RECORD FIELDS, THIS SECTION PRINTS THE INITIAL RECORD 10257
! DATA TO THE LOG FILE, MODIFIES THE RECORD AND PRINTS THE MODIFIED      10258
! RECORD DATA TO THE LOG FILE (HIGHLIGHTING THE CHANGED FIELDS)          10259
!=====                                                                  10260
!                                                                           10261
! IF TCONTINUE2 NE "A" AND                                                10262
! (SYSTEM_NAME NE TT$SYSNAME OR                                           10263
!   ITEM1 NE TTITEM1 OR ITEM2 NE TTITEM2 OR ITEM3 NE TTITEM3 OR          10264
!   ITEM4 NE TTITEM4 OR ITEM5 NE TTITEM5 OR ITEM6 NE TTITEM6 OR          10265
!   ITEM7 NE TTITEM7 OR ITEM8 NE TTITEM8 OR ITEM9 NE TTITEM9 OR          10266
!   ITEM10 NE TTITEM10 OR ITEM11 NE TTITEM11 OR ITEM12 NE TTITEM12 OR     10267
!   ITEM13 NE TTITEM13 OR ITEM14 NE TTITEM14 OR ITEM15 NE TTITEM15 OR     10268
!   REFERENCE1 NE TTREF1 OR REFERENCE2 NE TTREF2 OR                      10269
!   REFERENCE3 NE TTREF3 OR REFERENCE4 NE TTREF4 OR                      10270
!   REFERENCE5 NE TTREF5 OR REFERENCE6 NE TTREF6 OR                      10271
!   REFERENCE7 NE TTREF7 OR REFERENCE8 NE TTREF8 OR                      10272
!   REFERENCE9 NE TTREF9 OR REFERENCE10 NE TTREF10) THEN                 10273
! BEGIN                                                                    10274
!                                                                           10275
!=====                                                                  10276
! THIS SECTION PRINTS THE INITIAL RECORD DATA TO THE SESSION LOG         10277
! FILE                                                                     10278
!=====                                                                  10279
!                                                                           10280
!                                                                           10281
!                                                                           10282
!=====                                                                  10283
!                                                                           10284
! KCNT = KCNT + 1                                                         10285
! PRINT NEW_PAGE, SKIP 3,                                                 10286
!   COL 1, "RECORD NO. ", SPACE 0,                                       10287
!     KCNT (-) USING 9(4), SKIP 1,                                       10288
!     "===== ", SPACE 0,                                               10289
!     "===== ", SKIP 1,                                                10290
!     "===== ", SPACE 0,                                               10291
!     "===== ", SKIP 2,                                                10292
!   COL 6, "DATE_CREATED      :", SPACE 1,                               10293
!     DATE_CREATED (-) USING X(23), SKIP 1,                             10294
!   COL 6, "SYSTEM           :", SPACE 1,                               10295
!     SYSTEM (-) USING X(4), SKIP 1,                                     10296
!   COL 6, "SYSTEM_NAME      :", SPACE 1,                               10297
!     SYSTEM_NAME (-) USING T(52), SKIP 1,                             10298
!   COL 6, "FMEA ITEMS       :",                                         10299
!     COL 32, " 1)", SPACE 1, ITEM1 (-) USING X(4),                    10300

```


Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

COL 52, " 6)", SPACE 1, ITEM6 (-) USING X(4), 10301
COL 72, "11)", SPACE 1, ITEM11 (-) USING X(4), SKIP 1, 10302
COL 32, " 2)", SPACE 1, ITEM2 (-) USING X(4), 10303
COL 52, " 7)", SPACE 1, ITEM7 (-) USING X(4), 10304
COL 72, "12)", SPACE 1, ITEM12 (-) USING X(4), SKIP 1, 10305
COL 32, " 3)", SPACE 1, ITEM3 (-) USING X(4), 10306
COL 52, " 8)", SPACE 1, ITEM8 (-) USING X(4), 10307
COL 72, "13)", SPACE 1, ITEM13 (-) USING X(4), SKIP 1, 10308
COL 32, " 4)", SPACE 1, ITEM4 (-) USING X(4), 10309
COL 52, " 9)", SPACE 1, ITEM9 (-) USING X(4), 10310
COL 72, "14)", SPACE 1, ITEM14 (-) USING X(4), SKIP 1, 10311
COL 32, " 5)", SPACE 1, ITEM5 (-) USING X(4), 10312
COL 52, "10)", SPACE 1, ITEM10 (-) USING X(4), 10313
COL 72, "15)", SPACE 1, ITEM15 (-) USING X(4), SKIP 1, 10314
COL 6, "REFERENCES :", 10315
COL 32, " 1)", SPACE 1, REFERENCE1 (-) USING X(5), 10316
COL 52, " 5)", SPACE 1, REFERENCE5 (-) USING X(5), 10317
COL 72, " 9)", SPACE 1, REFERENCE9 (-) USING X(5), SKIP 1, 10318
COL 32, " 2)", SPACE 1, REFERENCE2 (-) USING X(5), 10319
COL 52, " 6)", SPACE 1, REFERENCE6 (-) USING X(5), 10320
COL 72, "10)", SPACE 1, REFERENCE10 (-) USING X(5), SKIP 1, 10321
COL 32, " 3)", SPACE 1, REFERENCE3 (-) USING X(5), 10322
COL 52, " 7)", SPACE 1, REFERENCE7 (-) USING X(5), SKIP 1, 10323
COL 32, " 4)", SPACE 1, REFERENCE4 (-) USING X(5), 10324
COL 52, " 8)", SPACE 1, REFERENCE8 (-) USING X(5), SKIP 1, 10325
COL 6, "PROPAGATIONS_FILE_", SKIP 1, 10326
COL 6, " CREATED :", SPACE 1, 10327
PROPAGATIONS_FILE_CREATED (-) USING X(3), SKIP 1, 10328
COL 6, "DATE_LAST_MODIFIED :", SPACE 1, 10329
DATE_LAST_MODIFIED (-) USING X(23), SKIP 1, 10330
COL 6, "MODIFYING_PROCEDURE :", SPACE 1, 10331
MODIFYING_PROCEDURE (-) USING X(20) 10332

```

```

=====
THIS SECTION STORES THE INITIAL RECORD VALUES FOR USE IN
HIGHLIGHTING THE FIELDS WHICH HAVE BEEN MODIFIED
=====

```

```

HDCREATED = DATE_CREATED 10341
HSYSTEM = SYSTEM 10342
HSYSNAME = SYSTEM_NAME 10343
HITEM1 = ITEM1 10344
HITEM2 = ITEM2 10345
HITEM3 = ITEM3 10346
HITEM4 = ITEM4 10347
HITEM5 = ITEM5 10348
HITEM6 = ITEM6 10349
HITEM7 = ITEM7 10350

```

!!!!!!

CAL = "NOW"		10382
MODIFY USING		10383
BEGIN		10384
SYSTEM_NAME	= TTSYSNAME	10385
ITEM1	= TTITEM1	10386
ITEM2	= TTITEM2	10387
ITEM3	= TTITEM3	10388
ITEM4	= TTITEM4	10389
ITEM5	= TTITEM5	10390
ITEM6	= TTITEM6	10391
ITEM7	= TTITEM7	10392
ITEM8	= TTITEM8	10393
ITEM9	= TTITEM9	10394
ITEM10	= TTITEM10	10395
ITEM11	= TTITEM11	10396
ITEM12	= TTITEM12	10397
ITEM13	= TTITEM13	10398
ITEM14	= TTITEM14	10399
ITEM15	= TTITEM15	10400
REFERENCE1	= TTREF1	10401
REFERENCE2	= TTREF2	10402

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

REFERENCE3      = TTREF3      10403
REFERENCE4      = TTREF4      10404
REFERENCE5      = TTREF5      10405
REFERENCE6      = TTREF6      10406
REFERENCE7      = TTREF7      10407
REFERENCE8      = TTREF8      10408
REFERENCE9      = TTREF9      10409
REFERENCE10     = TTREF10     10410
DATE_LAST_MODIFIED = CAL      10411
MODIFYING_PROCEDURE = "SYS_MODIFY" 10412
END                                                    10413

```

```

=====
THIS SECTION PRINTS THE MODIFIED RECORD DATA TO THE SESSION LOG
FILE AND HIGHLIGHTS THE FIELDS WHICH CONTAIN NEW INFORMATION
=====

```

```

FLG4 = "N"                                           10423
IF HITEM1 = ITEM1 AND HITEM2 = ITEM2 AND HITEM3 = ITEM3 AND 10424
  HITEM4 = ITEM4 AND HITEM5 = ITEM5 AND HITEM6 = ITEM6 AND 10425
  HITEM7 = ITEM7 AND HITEM8 = ITEM8 AND HITEM9 = ITEM9 AND 10426
  HITEM10 = ITEM10 AND HITEM11 = ITEM11 AND HITEM12 = ITEM12 AND 10427
  HITEM13 = ITEM13 AND HITEM14 = ITEM14 AND HITEM15 = ITEM15 THEN 10428
  FLG4 = "Y"                                           10429
FLG5 = "N"                                           10430
IF HREF1 = REFERENCE1 AND HREF2 = REFERENCE2 AND 10431
  HREF3 = REFERENCE3 AND HREF4 = REFERENCE4 AND 10432
  HREF5 = REFERENCE5 AND HREF6 = REFERENCE6 AND 10433
  HREF7 = REFERENCE7 AND HREF8 = REFERENCE8 AND 10434
  HREF9 = REFERENCE9 AND HREF10 = REFERENCE10 THEN 10435
  FLG5 = "Y"                                           10436
PRINT SKIP 1,                                         10437
  COL 1, "=====", SPACE 0, 10438
  "=====", SKIP 2, 10439
  COL 1, CHOICE 10440
    DATE_CREATED = HDCREATED THEN " " 10441
    ELSE "***" 10442
  END CHOICE, 10443
  SPACE 2, "DATE_CREATED :", SPACE 1, 10444
  DATE_CREATED (-) USING X(23), SKIP 1, 10445
  COL 1, CHOICE 10446
    SYSTEM = HSYSTEM THEN " " 10447
    ELSE "***" 10448
  END CHOICE, 10449
  SPACE 2, "SYSTEM :", SPACE 1, 10450
  SYSTEM (-) USING X(4), SKIP 1, 10451
  COL 1, CHOICE 10452
    SYSTEM_NAME = HSYSNAME THEN " " 10453

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

        ELSE "****"
        END CHOICE,
        SPACE 2, "SYSTEM_NAME      :", SPACE 1,
        SYSTEM_NAME (-) USING T(52), SKIP 1,
COL 1, CHOICE
        FLG4 = "Y" THEN "  "
        ELSE "****"
        END CHOICE,
        SPACE 2, "FMEA_ITEMS      :",
COL 29, CHOICE
        ITEM1 = HITEM1 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 0, " 1)", SPACE 1,
        ITEM1 (-) USING X(4),
COL 48, CHOICE
        ITEM6 = HITEM6 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 1, " 6)", SPACE 1,
        ITEM6 (-) USING X(4),
COL 68, CHOICE
        ITEM11 = HITEM11 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 1, "11)", SPACE 1,
        ITEM11 (-) USING X(4),
COL 29, CHOICE
        ITEM2 = HITEM2 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 0, " 2)", SPACE 1,
        ITEM2 (-) USING X(4),
COL 48, CHOICE
        ITEM7 = HITEM7 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 1, " 7)", SPACE 1,
        ITEM7 (-) USING X(4),
COL 68, CHOICE
        ITEM12 = HITEM12 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 1, "12)", SPACE 1,
        ITEM12 (-) USING X(4),
COL 29, CHOICE
        ITEM3 = HITEM3 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 0, " 3)", SPACE 1,
        ITEM3 (-) USING X(4),
COL 48, CHOICE
        ITEM8 = HITEM8 THEN "  "
        ELSE "****"
        END CHOICE, SPACE 1, " 8)", SPACE 1,
        ITEM8 (-) USING X(4),
COL 68, CHOICE
        ITEM13 = HITEM13 THEN "  "

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

        ELSE "***"
        END CHOICE, SPACE 1, "13)", SPACE 1,
        ITEM13 (-) USING X(4),
COL 29, CHOICE
        ITEM4 = HITEM4 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 0, " 4)", SPACE 1,
        ITEM4 (-) USING X(4),
COL 48, CHOICE
        ITEM9 = HITEM9 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 1, " 9)", SPACE 1,
        ITEM9 (-) USING X(4),
COL 68, CHOICE
        ITEM14 = HITEM14 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 1, "14)", SPACE 1,
        ITEM14 (-) USING X(4),
COL 29, CHOICE
        ITEM5 = HITEM5 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 0, " 5)", SPACE 1,
        ITEM5 (-) USING X(4),
COL 48, CHOICE
        ITEM10 = HITEM10 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 1, "10)", SPACE 1,
        ITEM10 (-) USING X(4),
COL 68, CHOICE
        ITEM15 = HITEM15 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 1, "15)", SPACE 1,
        ITEM15 (-) USING X(4),
COL 1, CHOICE
        FLG5 = "Y" THEN "   "
        ELSE "***"
        END CHOICE,
SPACE 2, "REFERENCES          :",
COL 29, CHOICE
        REFERENCE1 = HREF1 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 0, " 1)", SPACE 1,
        REFERENCE1 (-) USING X(5),
COL 48, CHOICE
        REFERENCE5 = HREF5 THEN "   "
        ELSE "***"
        END CHOICE, SPACE 1, " 5)", SPACE 1,
        REFERENCE5 (-) USING X(5),
COL 68, CHOICE
        REFERENCE9 = HREF9 THEN "   "
        ELSE "***"

```

Datatrieve Procedure SYS_MODIFY_2 (cont.)

END_CHOICE, SPACE 1, " 9)", SPACE 1,	10556
REFERENCE9 (-) USING X(5),	10557
COL 29, CHOICE	10558
REFERENCE2 = HREF2 THEN " "	10559
ELSE "****"	10560
END_CHOICE, SPACE 0, " 2)", SPACE 1,	10561
REFERENCE2 (-) USING X(5),	10562
COL 48, CHOICE	10563
REFERENCE6 = HREF6 THEN " "	10564
ELSE "****"	10565
END_CHOICE, SPACE 1, " 6)", SPACE 1,	10566
REFERENCE6 (-) USING X(5),	10567
COL 68, CHOICE	10568
REFERENCE10 = HREF10 THEN " "	10569
ELSE "****"	10570
END_CHOICE, SPACE 1, "10)", SPACE 1,	10571
REFERENCE10 (-) USING X(5),	10572
COL 29, CHOICE	10573
REFERENCE3 = HREF3 THEN " "	10574
ELSE "****"	10575
END_CHOICE, SPACE 0, " 3)", SPACE 1,	10576
REFERENCE3 (-) USING X(5),	10577
COL 48, CHOICE	10578
REFERENCE7 = HREF7 THEN " "	10579
ELSE "****"	10580
END_CHOICE, SPACE 1, " 7)", SPACE 1,	10581
REFERENCE7 (-) USING X(5),	10582
COL 29, CHOICE	10583
REFERENCE4 = HREF4 THEN " "	10584
ELSE "****"	10585
END_CHOICE, SPACE 0, " 4)", SPACE 1,	10586
REFERENCE4 (-) USING X(5),	10587
COL 48, CHOICE	10588
REFERENCE8 = HREF8 THEN " "	10589
ELSE "****"	10590
END_CHOICE, SPACE 1, " 8)", SPACE 1,	10591
REFERENCE8 (-) USING X(5),	10592
COL 1, CHOICE	10593
PROPAGATIONS_FILE_CREATED = HFIPCREATED THEN " "	10594
ELSE "****"	10595
END_CHOICE,	10596
SPACE 2, "PROPAGATIONS_FILE_ ", SKIP 1,	10597
COL 6, " CREATED :", SPACE 1,	10598
PROPAGATIONS_FILE_CREATED (-) USING X(3), SKIP 1,	10599
COL 1, CHOICE	10600
DATE_LAST_MODIFIED = HDLASTMOD THEN " "	10601
ELSE "****"	10602
END_CHOICE,	10603
SPACE 2, "DATE_LAST_MODIFIED :", SPACE 1,	10604
DATE_LAST_MODIFIED (-) USING X(23), SKIP 1,	10605
COL 1, CHOICE	10606

Datatrieve Procedure SYS_MODIFY_2 (cont.)

```

        MODIFYING_PROCEDURE = HMODPROC THEN "      "
        ELSE "****"
    END CHOICE,
    SPACE 2, "MODIFYING_PROCEDURE :",          SPACE 1,
        MODIFYING_PROCEDURE (-) USING X(20), SKIP 2,
COL 1, "===== ", SPACE 0,
        "===== ", SKIP 1,
        "===== ", SPACE 0,
        "===== "
: BELL
END
!
!
! =====
!
! IF TCONTINUE2 IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN MODIFIED. THIS
! MESSAGE WILL APPEAR IN THE SESSION LOG FILE.
!
! =====
!
IF TCONTINUE2 = "A" THEN
    BEGIN
        PRINT NEW_PAGE, SKIP 3,
        COL 1, "===== ", SPACE 0,
            "===== ", SKIP 12,
        COL 9, "*****", SKIP 2,
        COL 9, "***** RECORD NOT MODIFIED *****", SKIP 2,
        COL 9, "*****", SKIP 12,
        COL 1, "===== ", SPACE 0,
            "===== "
    END
END-PROCEDURE

```

Datatrieve Procedure SYS_STORE

```

DEFINE PROCEDURE SYS_STORE                                     10640
!                                                             10641
!                                                             10642
!=====                                                     10643
!                                                             10644
!  VARIABLES ASSOCIATED WITH SYSTEM AND SYSTEM_NAME          10645
!                                                             10646
!=====                                                     10647
DECLARE TSYSTEM      PIC X(4).                                10648
DECLARE TSYSNAME     PIC X(80).                               10649
!                                                             10650
!                                                             10651
!=====                                                     10652
!                                                             10653
!  VARIABLES ASSOCIATED WITH FMEA_ITEMS                       10654
!                                                             10655
!=====                                                     10656
!                                                             10657
DECLARE TITEM1       PIC X(4).                                10658
DECLARE TITEM2       PIC X(4).                                10659
DECLARE TITEM3       PIC X(4).                                10660
DECLARE TITEM4       PIC X(4).                                10661
DECLARE TITEM5       PIC X(4).                                10662
DECLARE TITEM6       PIC X(4).                                10663
DECLARE TITEM7       PIC X(4).                                10664
DECLARE TITEM8       PIC X(4).                                10665
DECLARE TITEM9       PIC X(4).                                10666
DECLARE TITEM10      PIC X(4).                                10667
DECLARE TITEM11      PIC X(4).                                10668
DECLARE TITEM12      PIC X(4).                                10669
DECLARE TITEM13      PIC X(4).                                10670
DECLARE TITEM14      PIC X(4).                                10671
DECLARE TITEM15      PIC X(4).                                10672
DECLARE TTITEM       PIC X(4).                                10673
!                                                             10674
!                                                             10675
!=====                                                     10676
!                                                             10677
!  VARIABLES ASSOCIATED WITH REFERENCES                       10678
!                                                             10679
!=====                                                     10680
!                                                             10681
DECLARE TREF1        PIC X(5).                                10682
DECLARE TREF2        PIC X(5).                                10683
DECLARE TREF3        PIC X(5).                                10684
DECLARE TREF4        PIC X(5).                                10685
DECLARE TREF5        PIC X(5).                                10686
DECLARE TREF6        PIC X(5).                                10687
DECLARE TREF7        PIC X(5).                                10688
DECLARE TREF8        PIC X(5).                                10689
DECLARE TREF9        PIC X(5).                                10690

```


Datatrieve Procedure SYS_STORE (cont.)

DECLARE TREF10	PIC X(5).	10691
DECLARE TTREF	PIC X(5).	10692
!		10693
!		10694
=====		10695
!		10696
!	VARIABLE ASSOCIATED WITH PROCESSING OF INPUTS FOR FMEA_ITEMS AND	10697
!	REFERENCES	10698
!		10699
=====		10700
!		10701
DECLARE TEMP	PIC X(60).	10702
!		10703
!		10704
=====		10705
!		10706
!	VARIABLES USED AS FLAGS OR CONDITION INDICATORS	10707
!		10708
=====		10709
!		10710
DECLARE TCONTINUE	PIC X(1).	10711
DECLARE MSG	PIC 9(2).	10712
DECLARE FLG1	PIC X(1).	10713
DECLARE FLG2	PIC X(1).	10714
!		10715
!		10716
=====		10717
!		10718
!	VARIABLES USED TO STORE MESSAGES FOR DISPLAY ON THE TDMS FORM	10719
!		10720
=====		10721
!		10722
DECLARE TMSG	PIC X(80).	10723
DECLARE TMSG1	PIC X(80).	10724
DECLARE TMSG2	PIC X(80).	10725
DECLARE TMSG2A	PIC X(80).	10726
DECLARE TMSG3	PIC X(80).	10727
DECLARE TMSG4	PIC X(80).	10728
DECLARE TMSG4A	PIC X(80).	10729
DECLARE TMSG5	PIC X(80).	10730
DECLARE TMSG5A	PIC X(80).	10731
DECLARE TMSG6	PIC X(80).	10732
!		10733
!		10734
=====		10735
!		10736
!	VARIABLES USED AS COUNTERS	10737
!		10738
=====		10739
!		10740
DECLARE ICNT	PIC 9(4).	10741

Datatrieve Procedure SYS_STORE (cont.)

DECLARE I	PIC 9(4).	10742
!		10743
!		10744
=====		10745
!		10746
!	VARIABLE USED TO OBTAIN CURRENT DATE AND TIME FROM DATATRIEVE	10747
!		10748
=====		10749
!		10750
DECLARE CAL	USAGE DATE	10751
	EDIT_STRING X(23).	10752
!		10753
!		10754
=====		10755
!		10756
!	READY THE APPROPRIATE DOMAINS, INITIALIZE THE MESSAGE VARIABLES AND	10757
!	INITIALIZE THE COUNTER (ICNT) USED FOR NUMBERING THE LOG FILE RECORDS	10758
!		10759
=====		10760
!		10761
SET ABORT		10762
READY SYSTEMS	SHARED WRITE	10763
READY REFERENCES	SHARED READ	10764
READY SYSTEMS FORM	SHARED READ	10765
TMSG1	= "ENTER DATA IN APPROPRIATE FIELDS AND PRESS RETURN KEY"	10766
TMSG2	= "SYSTEM IS NOT VALID --"	10767
TMSG2A	= "ALREADY EXISTS IN DOMAIN SYSTEMS"	10768
TMSG3	= "SYSTEM NAME IS NOT VALID -- ALREADY EXISTS IN DOMAIN SYSTEMS"	10769
TMSG4	= "FMEA ITEM IS NOT VALID --"	10770
TMSG4A	= "NOT IN FMEA_ITEM_NAME_TABLE"	10771
TMSG5	= "REFERENCE DOCUMENT IS NOT VALID --"	10772
TMSG5A	= "DOES NOT EXIST IN DOMAIN REFERENCES"	10773
TMSG6	= "CONTINUE IS NOT VALID -- MUST BE Y, N OR A"	10774
ICNT	= 0	10775
!		10776
!		10777
=====		10778
!		10779
!	PRIMARY LOOP TO STORE SYSTEMS	10780
!		10781
!	THIS LOOP ENCOMPASSES THE FOLLOWING FUNCTIONS:	10782
!	1. LOOP TO REQUEST INPUT DATA, TEST VALUES AND PROMPT FOR	10783
!	CORRECTION OF INVALID INFORMATION	10784
!	2. IF TCONTINUE NE "A", SECTION TO REARRANGE THE VARIABLES	10785
!	ASSOCIATED WITH FMEA ITEMS	10786
!	3. IF TCONTINUE NE "A", SECTION TO REARRANGE THE VARIABLES	10787
!	ASSOCIATED WITH REFERENCES	10788
!	4. IF TCONTINUE NE "A", SECTION TO STORE RECORD IN DOMAIN SYSTEMS	10789
!	AFTER VALIDATION TESTS HAVE BEEN PASSED	10790
!	5. IF TCONTINUE NE "A", SECTION TO PRINT DATA STORED IN DOMAIN	10791
!	SYSTEMS FOR INCLUSION IN THE SESSION LOG FILE	10792

Datatrieve Procedure SYS_STORE (cont.)

```

!      6. IF TCONTINUE = "A", SECTION TO PRINT MESSAGE THAT DATA CURRENTLY
!      ON FORM HAS NOT BEEN STORED
!      7. IF TCONTINUE = "A", SECTION TO REQUEST RESPONSE TO CONTINUE
!      PROCEDURE OR EXIT TO MENU
!=====
!
TCONTINUE = "Y"
WHILE TCONTINUE = "Y"
BEGIN
!=====
!      LOOP TO DISPLAY A BLANK TDMS FORM, RETRIEVE THE DATA ENTERED ON THE
!      FORM, TEST THE INPUT VALUES AND REQUEST CORRECTION OF INVALID DATA
!=====
!
FLG1 = "N"
IMSG = 1
WHILE FLG1 = "N"
BEGIN
    IF IMSG = 1 THEN TMSG = TMSG1
    IF IMSG = 2 THEN TMSG = TMSG2 || " " | TSYSTEM | " " | TMSG2A
    IF IMSG = 3 THEN TMSG = TMSG3
    IF IMSG = 4 THEN TMSG = TMSG4 || " " | TTITEM | " " | TMSG4A
    IF IMSG = 5 THEN TMSG = TMSG5 || " " | TTREF | " " | TMSG5A
    IF IMSG = 6 THEN TMSG = TMSG6
!=====
!      THIS SECTION DISPLAYS THE STORE SYSTEMS FORM AND RETRIEVES
!      THE DATA ENTERED ON THE FORM
!=====
!
FOR FIRST 1 SYSTEMS_FORM
BEGIN
    DISPLAY FORM SYSTEMS_STO FORM IN
    DEV$206:[BCDSSME2.FORMS]FORMSLIB.RLB USING
    BEGIN
        IF IMSG NE 1 THEN
            BEGIN
                PUT_FORM SYSTEM = TSYSTEM
                PUT_FORM SYSTEM_NAME = TSYSNAME
                PUT_FORM ITEM_1 = TITEM1
                PUT_FORM ITEM_2 = TITEM2
                PUT_FORM ITEM_3 = TITEM3
                PUT_FORM ITEM_4 = TITEM4

```

Datatrieve Procedure SYS_STORE (cont.)

PUT_FORM ITEM_5	= TITEM5	10844
PUT_FORM ITEM_6	= TITEM6	10845
PUT_FORM ITEM_7	= TITEM7	10846
PUT_FORM ITEM_8	= TITEM8	10847
PUT_FORM ITEM_9	= TITEM9	10848
PUT_FORM ITEM_10	= TITEM10	10849
PUT_FORM ITEM_11	= TITEM11	10850
PUT_FORM ITEM_12	= TITEM12	10851
PUT_FORM ITEM_13	= TITEM13	10852
PUT_FORM ITEM_14	= TITEM14	10853
PUT_FORM ITEM_15	= TITEM15	10854
PUT_FORM REF_1	= TREF1	10855
PUT_FORM REF_2	= TREF2	10856
PUT_FORM REF_3	= TREF3	10857
PUT_FORM REF_4	= TREF4	10858
PUT_FORM REF_5	= TREF5	10859
PUT_FORM REF_6	= TREF6	10860
PUT_FORM REF_7	= TREF7	10861
PUT_FORM REF_8	= TREF8	10862
PUT_FORM REF_9	= TREF9	10863
PUT_FORM REF_10	= TREF10	10864
END		10865
PUT_FORM CONTINUE = TCONTINUE		10866
PUT_FORM MESSAGE = TMSG		10867
END RETRIEVE USING		10868
BEGIN		10869
TSYSTEM = GET_FORM SYSTEM		10870
TSYSNAME = GET_FORM SYSTEM_NAME		10871
TITEM1 = GET_FORM ITEM_1		10872
TITEM2 = GET_FORM ITEM_2		10873
TITEM3 = GET_FORM ITEM_3		10874
TITEM4 = GET_FORM ITEM_4		10875
TITEM5 = GET_FORM ITEM_5		10876
TITEM6 = GET_FORM ITEM_6		10877
TITEM7 = GET_FORM ITEM_7		10878
TITEM8 = GET_FORM ITEM_8		10879
TITEM9 = GET_FORM ITEM_9		10880
TITEM10 = GET_FORM ITEM_10		10881
TITEM11 = GET_FORM ITEM_11		10882
TITEM12 = GET_FORM ITEM_12		10883
TITEM13 = GET_FORM ITEM_13		10884
TITEM14 = GET_FORM ITEM_14		10885
TITEM15 = GET_FORM ITEM_15		10886
TREF1 = GET_FORM REF_1		10887
TREF2 = GET_FORM REF_2		10888
TREF3 = GET_FORM REF_3		10889
TREF4 = GET_FORM REF_4		10890
TREF5 = GET_FORM REF_5		10891
TREF6 = GET_FORM REF_6		10892
TREF7 = GET_FORM REF_7		10893
TREF8 = GET_FORM REF_8		10894

Datatrieve Procedure SYS_STORE (cont.)

```

TREF9      = GET_FORM REF_9      10895
TREF10     = GET_FORM REF_10     10896
TCONTINUE  = GET_FORM CONTINUE  10897
END        10898
END        10899
!         10900
!         10901
!         10902
!         10903
!         10904
!         10905
!         10906
!         10907
!         10908
!         10909
!         10910
!         10911
!         10912
!         10913
!         10914
!         10915
!         10916
!         10917
!         10918
!         10919
!         10920
!         10921
!         10922
!         10923
!         10924
!         10925
!         10926
!         10927
!         10928
!         10929
!         10930
!         10931
!         10932
!         10933
!         10934
!         10935
!         10936
!         10937
!         10938
!         10939
!         10940
!         10941
!         10942
!         10943
!         10944
!         10945

=====
IF TCONTINUE IS NOT EQUAL TO "A", ANY LEADING BLANKS WHICH
WERE INADVERTENTLY ENTERED IN TSYSNAME ARE REMOVED
=====

IF TCONTINUE NE "A" THEN
BEGIN
    I = 1
    WHILE FN$STR_EXTRACT(TSYSNAME, I, 1) = " "
    BEGIN
        I = I + 1
    END
    TSYSNAME = FN$STR_EXTRACT(TSYSNAME, I, 80 - I + 1)
END

=====
IF TCONTINUE IS NOT EQUAL TO "A", TESTS ARE PERFORMED TO
VERIFY THE FOLLOWING:
1. TSYSTEM DOES NOT ALREADY EXIST IN DOMAIN SYSTEMS
2. TSYSNAME DOES NOT ALREADY EXIST IN DOMAIN SYSTEMS
3. ANY OF THE VARIABLES TITEM1 TO TITEM15 WHICH ARE NOT
   BLANK ARE IN FMEA ITEM NAME TABLE
4. ANY OF THE VARIABLES TREF1 TO TREF10 WHICH ARE NOT BLANK
   ARE IN DOMAIN REFERENCES
5. TCONTINUE IS "Y", "N" OR "A"
=====

FLG1 = "Y"
IF TCONTINUE NE "A" THEN
BEGIN
    FOR SYSTEMS WITH SYSTEM = TSYSTEM
    BEGIN
        FLG1 = "N"
        MSG = 2
    END
    IF FLG1 = "Y" THEN
    BEGIN
        FOR SYSTEMS WITH SYSTEM_NAME = TSYSNAME
        BEGIN

```

Datatrieve Procedure SYS_STORE (cont.)

```

                FLG1 = "N"                                10946
                MSG = 3                                10947
            END                                          10948
        END                                          10949
    IF FLG1 = "Y" THEN                                10950
        BEGIN                                          10951
            I = 1                                      10952
            WHILE FLG1 = "Y" AND I LE 15              10953
                BEGIN                                  10954
                    IF I = 1 THEN TTITEM = TITEM1      10955
                    IF I = 2 THEN TTITEM = TITEM2      10956
                    IF I = 3 THEN TTITEM = TITEM3      10957
                    IF I = 4 THEN TTITEM = TITEM4      10958
                    IF I = 5 THEN TTITEM = TITEM5      10959
                    IF I = 6 THEN TTITEM = TITEM6      10960
                    IF I = 7 THEN TTITEM = TITEM7      10961
                    IF I = 8 THEN TTITEM = TITEM8      10962
                    IF I = 9 THEN TTITEM = TITEM9      10963
                    IF I = 10 THEN TTITEM = TITEM10     10964
                    IF I = 11 THEN TTITEM = TITEM11    10965
                    IF I = 12 THEN TTITEM = TITEM12    10966
                    IF I = 13 THEN TTITEM = TITEM13    10967
                    IF I = 14 THEN TTITEM = TITEM14    10968
                    IF I = 15 THEN TTITEM = TITEM15    10969
                    IF TTITEM NE " " AND               10970
                        TTITEM NOT IN FMEA_ITEM_NAME_TABLE THEN 10971
                        BEGIN                            10972
                            FLG1 = "N"                  10973
                            MSG = 4                     10974
                        END                                10975
                    I = I + 1                            10976
                END                                          10977
            END                                          10978
        END                                          10979
    IF FLG1 = "Y" THEN                                10980
        BEGIN                                          10981
            I = 1                                      10982
            WHILE FLG1 = "Y" AND I LE 10              10983
                BEGIN                                  10984
                    IF I = 1 THEN TTREF = TREF1         10985
                    IF I = 2 THEN TTREF = TREF2         10986
                    IF I = 3 THEN TTREF = TREF3         10987
                    IF I = 4 THEN TTREF = TREF4         10988
                    IF I = 5 THEN TTREF = TREF5         10989
                    IF I = 6 THEN TTREF = TREF6         10990
                    IF I = 7 THEN TTREF = TREF7         10991
                    IF I = 8 THEN TTREF = TREF8         10992
                    IF I = 9 THEN TTREF = TREF9         10993
                    IF I = 10 THEN TTREF = TREF10        10994
                    IF TTREF NE " " THEN                10995
                        BEGIN                            10996
                            FLG2 = "N"

```

Datatrieve Procedure SYS_STORE (cont.)

```

FOR REFERENCES WITH REFERENCE_NUMBER = TTREF 10997
BEGIN 10998
    FLG2 = "Y" 10999
END 11000
IF FLG2 = "N" THEN 11001
    BEGIN 11002
        FLG1 = "N" 11003
        MSG = 5 11004
    END 11005
END 11006
    I = I + 1 11007
END 11008
END 11009
IF FLG1 = "Y" THEN 11010
    BEGIN 11011
        IF TCONTINUE NE "Y" AND 11012
           TCONTINUE NE "N" AND 11013
           TCONTINUE NE "A" THEN 11014
            BEGIN 11015
                FLG1 = "N" 11016
                MSG = 6 11017
            END 11018
        END 11019
    END 11020
END 11021
! 11022
! 11023
! 11024
! 11025
! 11026
! 11027
! 11028
! 11029
! 11030
! 11031
! 11032
! 11033
! 11034
! 11035
! 11036
! 11037
! 11038
! 11039
! 11040
! 11041
! 11042
! 11043
! 11044
! 11045
! 11046
! 11047

=====
IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE
VARIABLES TITEM1 TO TITEM15 SO THAT THE FIRST VALUE WHICH IS NOT
BLANK IS TITEM1, THE SECOND VALUE WHICH IS NOT BLANK IS TITEM2,
ETC.
=====

IF TCONTINUE NE "A" THEN
BEGIN
    TEMP = TITEM1 || TITEM2 || TITEM3 || TITEM4 || TITEM5 ||
           TITEM6 || TITEM7 || TITEM8 || TITEM9 || TITEM10 ||
           TITEM11 || TITEM12 || TITEM13 || TITEM14 || TITEM15 ||

    TITEM1 = FN$STR_EXTRACT(TEMP, 1, 4)
    TITEM2 = FN$STR_EXTRACT(TEMP, 5, 4)
    TITEM3 = FN$STR_EXTRACT(TEMP, 9, 4)
    TITEM4 = FN$STR_EXTRACT(TEMP, 13, 4)
    TITEM5 = FN$STR_EXTRACT(TEMP, 17, 4)
    TITEM6 = FN$STR_EXTRACT(TEMP, 21, 4)
    TITEM7 = FN$STR_EXTRACT(TEMP, 25, 4)
    TITEM8 = FN$STR_EXTRACT(TEMP, 29, 4)
    TITEM9 = FN$STR_EXTRACT(TEMP, 33, 4)
    TITEM10 = FN$STR_EXTRACT(TEMP, 37, 4)

```

Datatrieve Procedure SYS_STORE (cont.)

```

TITEM11 = FN$STR_EXTRACT(TEMP, 41, 4) 11048
TITEM12 = FN$STR_EXTRACT(TEMP, 45, 4) 11049
TITEM13 = FN$STR_EXTRACT(TEMP, 49, 4) 11050
TITEM14 = FN$STR_EXTRACT(TEMP, 53, 4) 11051
TITEM15 = FN$STR_EXTRACT(TEMP, 57, 4) 11052
END 11053
! 11054
! 11055
! 11056
! 11057
! 11058
! 11059
! 11060
! 11061
! 11062
! 11063
! 11064
! 11065
! 11066
! 11067
! 11068
! 11069
! 11070
! 11071
! 11072
! 11073
! 11074
! 11075
! 11076
! 11077
! 11078
! 11079
! 11080
! 11081
! 11082
! 11083
! 11084
! 11085
! 11086
! 11087
! 11088
! 11089
! 11090
! 11091
! 11092
! 11093
! 11094
! 11095
! 11096
! 11097
! 11098

```

IF TCONTINUE IS NOT EQUAL TO "A", THIS SECTION REARRANGES THE
VARIABLES TREF1 TO TREF10 SO THAT THE FIRST VALUE WHICH IS NOT
BLANK IS TREF1, THE SECOND VALUE WHICH IS NOT BLANK IS TREF2,
ETC.

```

IF TCONTINUE NE "A" THEN
BEGIN
TEMP = TREF1 || TREF2 || TREF3 || TREF4 || TREF5 ||
TREF6 || TREF7 || TREF8 || TREF9 || TREF10
TREF1 = FN$STR_EXTRACT(TEMP, 1, 5)
TREF2 = FN$STR_EXTRACT(TEMP, 6, 5)
TREF3 = FN$STR_EXTRACT(TEMP, 11, 5)
TREF4 = FN$STR_EXTRACT(TEMP, 16, 5)
TREF5 = FN$STR_EXTRACT(TEMP, 21, 5)
TREF6 = FN$STR_EXTRACT(TEMP, 26, 5)
TREF7 = FN$STR_EXTRACT(TEMP, 31, 5)
TREF8 = FN$STR_EXTRACT(TEMP, 36, 5)
TREF9 = FN$STR_EXTRACT(TEMP, 41, 5)
TREF10 = FN$STR_EXTRACT(TEMP, 46, 5)
END

```

IF TCONTINUE IS NOT EQUAL TO "A", THE VERIFIED DATA IS STORED IN
DOMAIN SYSTEMS

```

IF TCONTINUE NE "A" THEN
BEGIN
CAL = "NOW"
STORE SYSTEMS USING
BEGIN
DATE_CREATED = CAL
SYSTEM = TSYSTEM
SYSTEM_NAME = TSYSNAME
ITEM1 = TITEM1
ITEM2 = TITEM2

```


Datatrieve Procedure SYS_STORE (cont.)

```

ITEM3      = TITEM3      11099
ITEM4      = TITEM4      11100
ITEM5      = TITEM5      11101
ITEM6      = TITEM6      11102
ITEM7      = TITEM7      11103
ITEM8      = TITEM8      11104
ITEM9      = TITEM9      11105
ITEM10     = TITEM10     11106
ITEM11     = TITEM11     11107
ITEM12     = TITEM12     11108
ITEM13     = TITEM13     11109
ITEM14     = TITEM14     11110
ITEM15     = TITEM15     11111
REFERENCE1 = TREF1       11112
REFERENCE2 = TREF2       11113
REFERENCE3 = TREF3       11114
REFERENCE4 = TREF4       11115
REFERENCE5 = TREF5       11116
REFERENCE6 = TREF6       11117
REFERENCE7 = TREF7       11118
REFERENCE8 = TREF8       11119
REFERENCE9 = TREF9       11120
REFERENCE10 = TREF10     11121
PROPAGATIONS_FILE_CREATED = "NO" 11122
END      11123
END      11124
!      11125
!      11126
!      11127
!      11128
!      11129
!      11130
!      11131
!      11132
!      11133
!      11134
!      11135
!      11136
!      11137
!      11138
!      11139
!      11140
!      11141
!      11142
!      11143
!      11144
!      11145
!      11146
!      11147
!      11148
!      11149

=====
IF TCONTINUE IS NOT EQUAL TO "A", THE DATA STORED IN DOMAIN SYSTEMS
IS PRINTED.  THE OUTPUT OF THE PRINT STATEMENT WILL BE INCLUDED IN
THE SESSION LOG FILE WHICH IS OPENED BY THE CALLING COMMAND
PROCEDURE.
=====

IF TCONTINUE NE "A" THEN
BEGIN
  ICNT = ICNT + 1
  FOR SYSTEMS WITH DATE_CREATED = CAL      AND
                  SYSTEM      = TSYSTEM AND
                  SYSTEM NAME = TSYSNAME
  PRINT NEW_PAGE, SKIP 3,
    COL 1, "RECORD NO.",      SPACE 1,
    ICNT (-) USING ZZ9, SKIP 1,
    COL 1, "=====", SPACE 0,
    "=====", SPACE 0,
    "=====", SPACE 0,
    "=====", SKIP 2,
    COL 3, "DATE_CREATED :",      SPACE 1,

```

Datatrieve Procedure SYS_STORE (cont.)

	DATE CREATED (-) USING X(23), SKIP 1,	11150
COL 3,	"SYSTEM :", SPACE 1,	11151
	SYSTEM (-) USING X(4), SKIP 1,	11152
COL 3,	"SYSTEM_NAME :", SPACE 1,	11153
	SYSTEM_NAME (-) USING T(60), SKIP 1,	11154
COL 3,	"ITEM1 :", SPACE 1,	11155
	ITEM1 (-) USING X(4), SKIP 1,	11156
COL 3,	"ITEM2 :", SPACE 1,	11157
	ITEM2 (-) USING X(4), SKIP 1,	11158
COL 3,	"ITEM3 :", SPACE 1,	11159
	ITEM3 (-) USING X(4), SKIP 1,	11160
COL 3,	"ITEM4 :", SPACE 1,	11161
	ITEM4 (-) USING X(4), SKIP 1,	11162
COL 3,	"ITEM5 :", SPACE 1,	11163
	ITEM5 (-) USING X(4), SKIP 1,	11164
COL 3,	"ITEM6 :", SPACE 1,	11165
	ITEM6 (-) USING X(4), SKIP 1,	11166
COL 3,	"ITEM7 :", SPACE 1,	11167
	ITEM7 (-) USING X(4), SKIP 1,	11168
COL 3,	"ITEM8 :", SPACE 1,	11169
	ITEM8 (-) USING X(4), SKIP 1,	11170
COL 3,	"ITEM9 :", SPACE 1,	11171
	ITEM9 (-) USING X(4), SKIP 1,	11172
COL 3,	"ITEM10 :", SPACE 1,	11173
	ITEM10 (-) USING X(4), SKIP 1,	11174
COL 3,	"ITEM11 :", SPACE 1,	11175
	ITEM11 (-) USING X(4), SKIP 1,	11176
COL 3,	"ITEM12 :", SPACE 1,	11177
	ITEM12 (-) USING X(4), SKIP 1,	11178
COL 3,	"ITEM13 :", SPACE 1,	11179
	ITEM13 (-) USING X(4), SKIP 1,	11180
COL 3,	"ITEM14 :", SPACE 1,	11181
	ITEM14 (-) USING X(4), SKIP 1,	11182
COL 3,	"ITEM15 :", SPACE 1,	11183
	ITEM15 (-) USING X(4), SKIP 1,	11184
COL 3,	"REFERENCE1 :", SPACE 1,	11185
	REFERENCE1 (-) USING X(5), SKIP 1,	11186
COL 3,	"REFERENCE2 :", SPACE 1,	11187
	REFERENCE2 (-) USING X(5), SKIP 1,	11188
COL 3,	"REFERENCE3 :", SPACE 1,	11189
	REFERENCE3 (-) USING X(5), SKIP 1,	11190
COL 3,	"REFERENCE4 :", SPACE 1,	11191
	REFERENCE4 (-) USING X(5), SKIP 1,	11192
COL 3,	"REFERENCE5 :", SPACE 1,	11193
	REFERENCE5 (-) USING X(5), SKIP 1,	11194
COL 3,	"REFERENCE6 :", SPACE 1,	11195
	REFERENCE6 (-) USING X(5), SKIP 1,	11196
COL 3,	"REFERENCE7 :", SPACE 1,	11197
	REFERENCE7 (-) USING X(5), SKIP 1,	11198
COL 3,	"REFERENCE8 :", SPACE 1,	11199
	REFERENCE8 (-) USING X(5), SKIP 1,	11200

Datatrieve Procedure SYS_STORE (cont.)

```

COL 3, "REFERENCE9 :", SPACE 1, 11201
REFERENCE9 (-) USING X(5), SKIP 1, 11202
COL 3, "REFERENCE10 :", SPACE 1, 11203
REFERENCE10 (-) USING X(5), SKIP 2, 11204
COL 1, "=====", SPACE 0, 11205
"=====", SPACE 0, 11206
"=====", SPACE 0, 11207
"===== 11208

:BELL 11209
END 11210
! 11211
! 11212
! 11213
! ===== 11214
! IF TCONTINUE IS EQUAL TO "A", A MESSAGE IS PRINTED TO INDICATE THAT 11215
! THE DATA CURRENTLY SHOWN ON THE FORM HAS NOT BEEN STORED. THIS 11216
! MESSAGE WILL ALSO APPEAR IN THE SESSION LOG FILE. 11217
! ===== 11218
! 11219
! 11220
IF TCONTINUE = "A" THEN 11221
BEGIN 11222
PRINT NEW_PAGE, SKIP 3, 11223
COL 1, "===== ", SPACE 0, 11224
"===== ", SKIP 12, 11225
COL 9, "*****", SKIP 2, 11226
COL 9, "***** RECORD NOT STORED *****", SKIP 2, 11227
COL 9, "*****", SKIP 12, 11228
COL 1, "===== ", SPACE 0, 11229
"===== 11230

END 11231
! 11232
! 11233
! 11234
! ===== 11235
! IF TCONTINUE EQUALS "A", A RESPONSE IS REQUESTED TO EITHER CONTINUE 11236
! THE PROCEDURE TO STORE SYSTEMS OR EXIT TO THE MENU 11237
! ===== 11238
! 11239
! 11240
IF TCONTINUE = "A" THEN 11241
BEGIN 11242
PRINT NEW_PAGE 11243
:CLRSCRN 11244
TCONTINUE = "X" 11245
WHILE TCONTINUE NE "Y" AND 11246
TCONTINUE NE "N" 11247
BEGIN 11248
PRINT SKIP 2, 11249
"Do you wish to continue entering SYSTEMS?", 11250
SKIP 1 11251

```

Datatrieve Procedure SYS_STORE (cont.)

TCONTINUE = FN\$UPCASE(*."Y or N")	11252
PRINT " "	11253
END	11254
END	11255
END	11256
!	11257
!	11258
=====	11259
!	11260
! THIS SECTION CALLS PROCEDURE CREATE_PROPAGATIONS_SYS_1 TO CREATE A	11261
! PROPAGATIONS DOMAIN AND DATA FILE FOR EACH SYSTEM ENTERED.	11262
!	11263
=====	11264
!	11265
:CREATE_PROPAGATIONS_SYS_1	11266
FN\$DELETE_LOG("PROC")	11267
END-PROCEDURE	11268

F-260

(This page intentionally blank)

APPENDIX G

FIPM DATATRIEVE TABLES

<u>Datatrieve Table</u>	<u>Page</u>
ACCOMPLICE_REQUIRED_TABLE	G-3
CONNECTION_TABLE	G-4
FAILURE_MODE_SUBMODE_TABLE	G-5
FMEA_ITEM_NAME_TABLE	G-6
FMEA_ITEM_PART_NO_TABLE	G-15
FREQ_TIME_UNITS_TABLE	G-24
MONTH_TABLE	G-25
NUMBER_TABLE	G-26
PARAMETER_TABLE	G-27
REFERENCE_ABBREV_TABLE	G-28
REFERENCE_SOURCE_TABLE	G-29
SIGNAL_TABLE	G-30
SIGN_TABLE	G-31

G-2

(This page intentionally blank)

Datatrieve Table ACCOMPLICE_REQUIRED_TABLE

DEFINE TABLE ACCOMPLICE_REQUIRED_TABLE	0001
!	0002
!	0003
"LKN" : "LEAK: CONNECTION"	0004
!	0005
"WRRB" : "WEAR: RUBBING"	0006
!	0007
!	0008
END_TABLE	0009

PRECEDING PAGE BLANK NOT FILMED

Datatrieve Table CONNECTION_TABLE

DEFINE TABLE CONNECTION_TABLE	0010
!	0011
!	0012
"GAH2" : "GASEOUS: HYDROGEN"	0013
!	0014
"GAHE" : "GASEOUS: HELIUM"	0015
!	0016
"GAHG" : "GASEOUS: HOT GAS"	0017
!	0018
"GAO2" : "GASEOUS: OXYGEN"	0019
!	0020
"LQH2" : "LIQUID: HYDROGEN"	0021
!	0022
"LQHE" : "LIQUID: HELIUM"	0023
!	0024
"LQO2" : "LIQUID: OXYGEN"	0025
!	0026
"ME--" : "MECHANICAL"	0027
!	0028
"MECP" : "MECHANICAL: COMMON PIECE"	0029
!	0030
"MERE" : "MECHANICAL: ROLLING ELEMENT"	0031
!	0032
"TPH2" : "TWO PHASE: HYDROGEN"	0033
!	0034
"TPHE" : "TWO PHASE: HELIUM"	0035
!	0036
"TPO2" : "TWO PHASE: OXYGEN"	0037
!	0038
!	0039
END_TABLE	0040

Datatrieve Table FAILURE_MODE_SUBMODE_TABLE

DEFINE TABLE FAILURE_MODE_SUBMODE_TABLE	0041
!	0042
!	0043
"CROX" : "CORROSION: OXIDATION"	0044
!	0045
"DFIM" : "DEFORMATION: IMPACT"	0046
!	0047
"DFIP" : "DEFORMATION: INTERNAL PRESSURE"	0048
!	0049
"DFST" : "DEFORMATION: STATIC LOADING"	0050
!	0051
"DFSD" : "DEFORMATION: SURFACE DEPOSITION"	0052
!	0053
"FAIM" : "FRACTURE: IMPACT"	0054
!	0055
"FAIP" : "FRACTURE: INTERNAL PRESSURE"	0056
!	0057
"FAST" : "FRACTURE: STATIC LOADING"	0058
!	0059
"FATF" : "FRACTURE: THERMAL FATIGUE"	0060
!	0061
"FAVF" : "FRACTURE: VIBRATION FATIGUE"	0062
!	0063
"FIBN" : "FRICTION: BINDING"	0064
!	0065
"FISL" : "FRICTION: SLIPPAGE"	0066
!	0067
"LKCN" : "LEAK: CONNECTION"	0068
!	0069
"LKER" : "LEAK: EROSION"	0070
!	0071
"LKFA" : "LEAK: FRACTURE"	0072
!	0073
"LKPD" : "LEAK: PRESSURE DIFFERENTIAL"	0074
!	0075
"LKTL" : "LEAK: TOLERANCE"	0076
!	0077
"MPSD" : "MATERIAL PROPERTIES: SURFACE DEPOSITION"	0078
!	0079
"WRCV" : "WEAR: CAVITATION"	0080
!	0081
"WRER" : "WEAR: EROSION"	0082
!	0083
"WRPT" : "WEAR: PITTING"	0084
!	0085
"WRRE" : "WEAR: ROLLING ELEMENTS"	0086
!	0087
"WRRB" : "WEAR: RUBBING"	0088
!	0089
!	0090
END_TABLE	0091

Datatrieve Table FMEA_ITEM_NAME_TABLE

DEFINE TABLE FMEA_ITEM_NAME_TABLE	0092
EDIT_STRING IS T(80)	0093
!	0094
!	0095
"A100" : "HOT-GAS MANIFOLD"	0096
!	0097
"A150" : "HEAT EXCHANGER"	0098
!	0099
"A200" : "MAIN INJECTOR"	0100
!	0101
"A330" : "MAIN CHAMBER"	0102
!	0103
"A340" : "MAIN NOZZLE"	0104
!	0105
"A600" : "FUEL PREBURNER"	0106
!	0107
"A700" : "OXIDIZER PREBURNER"	0108
!	0109
"B200" : "HIGH-PRESSURE FUEL TURBOPUMP"	0110
!	0111
"B400" : "HIGH-PRESSURE OXIDIZER TURBOPUMP"	0112
!	0113
"B600" : "LOW-PRESSURE FUEL TURBOPUMP"	0114
!	0115
"B800" : "LOW-PRESSURE OXIDIZER TURBOPUMP"	0116
!	0117
"C111" : "FUEL PREBURNER PURGE CHECK VALVE"	0118
!	0119
"C112" : "OXIDIZER PREBURNER PURGE CHECK VALVE"	0120
!	0121
"C113" : "OXIDIZER DOME PURGE CHECK VALVE"	0122
!	0123
"C114" : "FUEL PURGE CHECK VALVE"	0124
!	0125
"C116" : "FUEL PREBURNER AUGMENTED SPARK IGNITER PURGE CHECK VALVE"	0126
!	0127
"C117" : "OXIDIZER PREBURNER AUGMENTED SPARK IGNITER PURGE CHECK VALVE"	0128
!	0129
"C200" : "PNEUMATIC CONTROL ASSEMBLY"	0130
!	0131
"C212" : "BLEED SOLENOID VALVE"	0132
!	0133
"C213" : "FUEL SYSTEM PURGE SOLENOID VALVE"	0134
!	0135
"C214" : "SHUTDOWN SOLENOID VALVE"	0136
!	0137
"C215" : "HIGH-PRESSURE OXIDIZER TURBOPUMP INTERMEDIATE PURGE SOLENOID VALVE"	0138
!	0139
"C217" : "PRECHARGE SOLENOID VALVE"	0140
!	0141
"C218" : "PREBURNER SHUTDOWN PURGE SOLENOID VALVE"	0142

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

!		0143
"C251"	: "FUEL SYSTEM PURGE PRESSURE-ACTIVATED VALVE"	0144
!		0145
"C252"	: "MAIN COMBUSTION CHAMBER DOME PRESSURE-ACTIVATED VALVE"	0146
!		0147
"C254"	: "OXIDIZER BLEED PRESSURE-ACTIVATED VALVE"	0148
!		0149
"C255"	: "SHUTDOWN PRESSURE-ACTIVATED VALVE"	0150
!		0151
"C257"	: "FUEL PREBURNER/AUGMENTED SPARK IGNITER PRESSURE-ACTIVATED VALVE"	0152
!		0153
"C258"	: "OXIDIZER PREBURNER/AUGMENTED SPARK IGNITER PRESSURE-ACTIVATED VALVE"	0154
!		0155
"C259"	: "PREBURNER SHUTDOWN PURGE SHUTOFF PRESSURE-ACTIVATED VALVE"	0156
!		0157
"C260"	:	0158
"HIGH-PRESSURE OXIDIZER TURBOPUMP INTERMEDIATE SEAL PRESSURE-ACTIVATED VALVE"		0159
!		0160
"C300"	: "HELIUM PRECHARGE VALVE ASSEMBLY"	0161
!		0162
"D110"	: "MAIN FUEL VALVE"	0163
!		0164
"D120"	: "MAIN OXIDIZER VALVE"	0165
!		0166
"D130"	: "FUEL PREBURNER OXIDIZER VALVE"	0167
!		0168
"D140"	: "OXIDIZER PREBURNER OXIDIZER VALVE"	0169
!		0170
"D150"	: "CHAMBER COOLANT VALVE"	0171
!		0172
"D210"	: "FUEL BLEED VALVE"	0173
!		0174
"D220"	: "OXIDIZER BLEED VALVE"	0175
!		0176
"D300"	: "ANTIFLOOD VALVE"	0177
!		0178
"D500"	: "GASEOUS OXYGEN CONTROL VALVE"	0179
!		0180
"D600"	: "RECIRCULATION ISOLATION VALVE"	0181
!		0182
"E110"	: "MAIN FUEL VALVE ACTUATOR"	0183
!		0184
"E120"	: "MAIN OXIDIZER VALVE ACTUATOR"	0185
!		0186
"E130"	: "FUEL PREBURNER OXIDIZER VALVE ACTUATOR"	0187
!		0188
"E140"	: "OXIDIZER PREBURNER OXIDIZER VALVE ACTUATOR"	0189
!		0190
"E150"	: "CHAMBER COOLANT VALVE ACTUATOR"	0191
!		0192
"F000"	: "CONTROLLER"	0193

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

!		0194
"F800" :	"FLIGHT ACCELERATION SAFETY CUTOFF SYSTEM"	0195
!		0196
"G100" :	"MAIN IGNITER NUMBER 1"	0197
!		0198
"G200" :	"MAIN IGNITER NUMBER 2"	0199
!		0200
"G300" :	"FUEL PREBURNER IGNITER NUMBER 1"	0201
!		0202
"G400" :	"FUEL PREBURNER IGNITER NUMBER 2"	0203
!		0204
"G500" :	"OXIDIZER PREBURNER IGNITER NUMBER 1"	0205
!		0206
"G600" :	"OXIDIZER PREBURNER IGNITER NUMBER 2"	0207
!		0208
"H101" :	"ELECTRICAL HARNESS 1W1"	0209
!		0210
"H102" :	"ELECTRICAL HARNESS 1W2"	0211
!		0212
"H103" :	"ELECTRICAL HARNESS 1W3"	0213
!		0214
"H104" :	"ELECTRICAL HARNESS 1W4"	0215
!		0216
"H105" :	"ELECTRICAL HARNESS 1W5"	0217
!		0218
"H107" :	"ELECTRICAL HARNESS 1W7"	0219
!		0220
"H108" :	"ELECTRICAL HARNESS 1W8"	0221
!		0222
"H109" :	"ELECTRICAL HARNESS 1W9"	0223
!		0224
"H110" :	"ELECTRICAL HARNESS 1W10"	0225
!		0226
"H111" :	"ELECTRICAL HARNESS 1W11"	0227
!		0228
"H112" :	"ELECTRICAL HARNESS 1W12"	0229
!		0230
"H113" :	"ELECTRICAL HARNESS 1W13"	0231
!		0232
"H114" :	"ELECTRICAL HARNESS 1W14"	0233
!		0234
"H115" :	"ELECTRICAL HARNESS 1W15"	0235
!		0236
"H116" :	"ELECTRICAL HARNESS 1W16"	0237
!		0238
"H118" :	"ELECTRICAL HARNESS 1W18"	0239
!		0240
"H119" :	"ELECTRICAL HARNESS 1W19"	0241
!		0242
"H120" :	"ELECTRICAL HARNESS 1W20"	0243
!		0244

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

"H122" : "ELECTRICAL HARNESS 1W22"	0245
!	0246
"H128" : "ELECTRICAL HARNESS 1W106"	0247
!	0248
"J201" : "MAIN CHAMBER PRESSURE SENSOR NUMBER 1"	0249
!	0250
"J202" : "MAIN CHAMBER PRESSURE SENSOR NUMBER 2"	0251
!	0252
"J203" : "LOW-PRESSURE OXIDIZER TURBOPUMP DISCHARGE PRESSURE SENSOR"	0253
!	0254
"J205" : "FUEL PREBURNER CHAMBER PRESSURE SENSOR"	0255
!	0256
"J207" : "OXIDIZER TANK PRESSURE SENSOR"	0257
!	0258
"J208" : "HIGH-PRESSURE FUEL TURBOPUMP DISCHARGE PRESSURE SENSOR"	0259
!	0260
"J209" : "HIGH-PRESSURE OXIDIZER TURBOPUMP BOOST DISCHARGE PRESSURE SENSOR"	0261
!	0262
"J210" : "FUEL INJECTION PRESSURE SENSOR"	0263
!	0264
"J211" : "HYDRAULIC PRESSURE SENSOR"	0265
!	0266
"J213" : "FUEL PURGE PRESSURE SENSOR"	0267
!	0268
"J216" :	0269
"HIGH-PRESSURE OXIDIZER TURBOPUMP INTERMEDIATE SEAL PURGE PRESSURE SENSOR"	0270
!	0271
"J218" : "LOW-PRESSURE FUEL TURBOPUMP DISCHARGE PRESSURE SENSOR"	0272
!	0273
"J220" : "HIGH-PRESSURE OXIDIZER TURBOPUMP DISCHARGE PRESSURE SENSOR"	0274
!	0275
"J221" : "COOLANT OUTLET PRESSURE SENSOR"	0276
!	0277
"J222" : "PRECHARGE PRESSURE SENSOR"	0278
!	0279
"J223" : "FUEL PREBURNER/AUGMENTED SPARK IGNITER PURGE PRESSURE SENSOR"	0280
!	0281
"J224" : "OXIDIZER PREBURNER/AUGMENTED SPARK IGNITER PURGE PRESSURE SENSOR"	0282
!	0283
"J225" : "PNEUMATIC SHUTDOWN SUPPLY PRESSURE SENSOR"	0284
!	0285
"J228" :	0286
"HIGH-PRESSURE OXIDIZER TURBOPUMP SECONDARY SEAL CAVITY PRESSURE SENSOR"	0287
!	0288
"J229" : "HIGH-PRESSURE OXIDIZER TURBOPUMP COOLANT LINER PRESSURE SENSOR"	0289
!	0290
"J301" : "HIGH-PRESSURE FUEL TURBOPUMP DISCHARGE TEMPERATURE SENSOR NUMBER 1"	0291
!	0292
"J302" : "HIGH-PRESSURE FUEL TURBOPUMP DISCHARGE TEMPERATURE SENSOR NUMBER 2"	0293
!	0294
"J303" :	0295

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

"HIGH-PRESSURE OXIDIZER TURBOPUMP DISCHARGE TEMPERATURE SENSOR NUMBER 1"	0296
!	0297
"J304" :	0298
"HIGH-PRESSURE OXIDIZER TURBOPUMP DISCHARGE TEMPERATURE SENSOR NUMBER 2"	0299
!	0300
"J306" : "LOW-PRESSURE FUEL TURBOPUMP DISCHARGE TEMPERATURE SENSOR"	0301
!	0302
"J309" : "COOLANT OUTLET TEMPERATURE SENSOR"	0303
!	0304
"J310" : "MAIN OXIDIZER VALVE WARMANT TEMPERATURE SENSOR"	0305
!	0306
"J311" : "MAIN FUEL VALVE WARMANT TEMPERATURE SENSOR"	0307
!	0308
"J312" : "HIGH-PRESSURE OXIDIZER TURBOPUMP BOOST DISCHARGE TEMPERATURE SENSOR"	0309
!	0310
"J601" : "FUEL FLOW PICKUP NUMBER 1"	0311
!	0312
"J602" : "FUEL FLOW PICKUP NUMBER 2"	0313
!	0314
"J607" : "LOW-PRESSURE FUEL TURBOPUMP SPEED PICKUP"	0315
!	0316
"J608" : "HIGH-PRESSURE FUEL TURBOPUMP SPEED PICKUP"	0317
!	0318
"J701" : "FUEL FLOWMETER"	0319
!	0320
"K101" : "LOW-PRESSURE FUEL TURBOPUMP DISCHARGE DUCT"	0321
!	0322
"K102" : "LOW-PRESSURE FUEL TURBOPUMP TURBINE DRIVE DUCT"	0323
!	0324
"K103" : "LOW-PRESSURE FUEL TURBOPUMP TURBINE DISCHARGE DUCT"	0325
!	0326
"K104" : "FUEL BLEED DUCT"	0327
!	0328
"K105" : "PREBURNER FUEL SUPPLY DUCT"	0329
!	0330
"K106" : "HIGH-PRESSURE FUEL DUCT"	0331
!	0332
"K107" : "FUEL TANK PRESSURE DUCT"	0333
!	0334
"K108" : "HOT-GAS MANIFOLD COOLANT DUCT"	0335
!	0336
"K110" : "FUEL BLEED DUCT"	0337
!	0338
"K111" : "AUGMENTED SPARK IGNITER FUEL SUPPLY DUCT"	0339
!	0340
"K112" : "OXIDIZER PREBURNER AUGMENTED SPARK IGNITER FUEL SUPPLY DUCT"	0341
!	0342
"K113" : "FUEL PREBURNER AUGMENTED SPARK IGNITER FUEL SUPPLY DUCT"	0343
!	0344
"K201" : "LOW-PRESSURE OXIDIZER TURBOPUMP DISCHARGE DUCT"	0345
!	0346

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

"K202" : "LOW-PRESSURE OXIDIZER TURBOPUMP TURBINE DRIVE DUCT"	0347
!	0348
"K203" : "OXIDIZER BLEED VALVE"	0349
!	0350
"K204" : "OXIDIZER TANK PRESSURE DUCT"	0351
!	0352
"K205" : "HIGH-PRESSURE OXIDIZER DUCT"	0353
!	0354
"K206" : "FUEL PREBURNER OXIDIZER SUPPLY DUCT"	0355
!	0356
"K207" : "HEAT EXCHANGER SUPPLY DUCT"	0357
!	0358
"K208" : "PREBURNER PUMP INLET DUCT"	0359
!	0360
"K209" : "THRUST CHAMBER AUGMENTED SPARK IGNITOR OXIDIZER SUPPLY LINE"	0361
!	0362
"K210" : "FUEL PREBURNER AUGMENTED SPARK IGNITOR OXIDIZER SUPPLY LINE"	0363
!	0364
"K211" : "OXIDIZER PREBURNER AUGMENTED SPARK IGNITOR OXIDIZER SUPPLY LINE"	0365
!	0366
"K212" : "OXIDIZER PREBURNER OXIDIZER SUPPLY DUCT"	0367
!	0368
"K213" : "OXIDIZER BLEED LINE"	0369
!	0370
"K214" : "OXIDIZER RECIRCULATION BLEED LINE"	0371
!	0372
"K215" : "POGO GASEOUS OXYGEN SUPPLY LINE"	0373
!	0374
"K216" : "RECIRCULATION ISOLATION VALVE OVERRIDE LINE"	0375
!	0376
"K217" : "ACCUMULATOR SUPPLY LINE"	0377
!	0378
"K218" : "POGO PRECHARGE LINE"	0379
!	0380
"K301" : "HYDRAULIC DRAIN LINES"	0381
!	0382
"K302" : "FUEL DRAIN LINES"	0383
!	0384
"K303" : "OXIDIZER DRAIN LINES"	0385
!	0386
"K304" : "HIGH-PRESSURE OXIDIZER TURBOPUMP PRIMARY TURBINE SEAL DRAIN LINE"	0387
!	0388
"K305" : "HIGH-PRESSURE OXIDIZER TURBOPUMP SECONDARY TURBINE SEAL DRAIN LINE"	0389
!	0390
"K306" : "HIGH-PRESSURE OXIDIZER SEAL DRAIN LINE"	0391
!	0392
"K401" : "HYDRAULIC SUPPLY HOSE"	0393
!	0394
"K402" : "HYDRAULIC SUPPLY LINE"	0395
!	0396
"K403" : "HYDRAULIC RETURN HOSE"	0397

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

!		0398
"K501" :	"HELIUM SUPPLY LINE"	0399
!		0400
"K502" :	"NITROGEN SUPPLY LINE"	0401
!		0402
"K504" :	"FUEL PURGE LINE"	0403
!		0404
"K505" :	"INTERMEDIATE SEAL PURGE LINE"	0405
!		0406
"K506" :	"FUEL BLEED VALVE CONTROL LINE NUMBER 1"	0407
!		0408
"K507" :	"OXIDIZER BLEED VALVE CONTROL LINE"	0409
!		0410
"K508" :	"OXIDIZER PREBURNER OXIDIZER VALVE PNEUMATIC SHUTDOWN LINE"	0411
!		0412
"K509" :	"MAIN CHAMBER OXIDIZER PURGE LINE NUMBER 1"	0413
!		0414
"K510" :	"HIGH-PRESSURE OXIDIZER TURBOPUMP TURBINE SEAL PURGE LINE"	0415
!		0416
"K511" :		0417
"MAIN OXIDIZER VALVE/FUEL PREBURNER OXIDIZER VALVE PNEUMATIC SHUTDOWN LINE"		0418
!		0419
"K513" :	"MAIN FUEL VALVE/CHAMBER COOLANT VALVE PNEUMATIC SHUTDOWN LINE"	0420
!		0421
"K514" :	"CHAMBER COOLANT VALVE PNEUMATIC SEQUENCE LINE"	0422
!		0423
"K515" :	"HELIUM SUPPLY LINE NUMBER 1"	0424
!		0425
"K516" :	"NITROGEN SUPPLY LINE"	0426
!		0427
"K517" :	"OXIDIZER PREBURNER OXIDIZER PURGE LINE"	0428
!		0429
"K518" :	"PREBURNER VALVE OXIDIZER PURGE LINE"	0430
!		0431
"K519" :	"HELIUM SUPPLY LINE NUMBER 2"	0432
!		0433
"K520" :	"FUEL BLEED VALVE CONTROL LINE NUMBER 2"	0434
!		0435
"K530" :	"MAIN CHAMBER OXIDIZER PURGE LINE NUMBER 2"	0436
!		0437
"K531" :	"POGO HELIUM VENT LINE"	0438
!		0439
"K532" :	"POGO HELIUM SUPPLY LINE"	0440
!		0441
"K533" :	"POGO GASEOUS OXYGEN CONTROL LINE"	0442
!		0443
"K534" :	"RECIRCULATION ISOLATION VALVE CONTROL LINE"	0444
!		0445
"K600" :	"CONTROLLER COOLING DUCT"	0446
!		0447
"L101" :	"FUEL SEALS"	0448

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

!		0449
"L102" :	"OXIDIZER SEALS"	0450
!		0451
"L103" :	"HOT-GAS SEALS"	0452
!		0453
"L104" :	"HYDRAULIC SEALS"	0454
!		0455
"L105" :	"PNEUMATIC SEALS"	0456
!		0457
"M000" :	"GIMBAL"	0458
!		0459
"N300" :	"ENGINE VEHICLE INTERFACE"	0460
!		0461
"N400" :	"POGO ACCUMULATOR"	0462
!		0463
"N701" :	"PNEUMATIC CONTROL ASSEMBLY FUEL SYSTEM PURGE ORIFICE (P3)"	0464
!		0465
"N702" :	"COMPONENT FUEL DRAIN ORIFICE (DIS. 3)"	0466
!		0467
"N703" :	"OXIDIZER PREBURNER OXIDIZER VALVE OXIDIZER SYSTEM PURGE ORIFICE (N5)"	0468
!		0469
"N704" :	"FUEL PREBURNER OXIDIZER SYSTEM PURGE ORIFICE (N8)"	0470
!		0471
"N705" :	"OXIDIZER PREBURNER OXIDIZER SYSTEM PURGE ORIFICE (N9)"	0472
!		0473
"N706" :	"HIGH-PRESSURE FUEL TURBOPUMP BEARING PURGE ORIFICE (N11)"	0474
!		0475
"N707" :	"MAIN INJECTOR OXIDIZER PURGE ORIFICE (N7)"	0476
!		0477
"N708" :	"MAIN COMBUSTION CHAMBER DRYING PURGE ORIFICE (N15)"	0478
!		0479
"N709" :	"GASEOUS OXYGEN CONTROL VALVE GASEOUS OXYGEN OUTLET ORIFICE (026)"	0480
!		0481
"N710" :	"HEAT EXCHANGER BYPASS ORIFICE (019.1)"	0482
!		0483
"N711" :		0484
"FUEL PREBURNER AUGMENTED SPARK IGNITER OXIDIZER ORIFICE (LINE RS007187)"		0485
!		0486
"N712" :		0487
"OXIDIZER PREBURNER AUGMENTED SPARK IGNITER OXIDIZER ORIFICE (LINE RS007186)"		0488
!		0489
"N713" :		0490
"OXIDIZER PREBURNER AUGMENTED SPARK IGNITER OXIDIZER ORIFICE (012.1.3)"		0491
!		0492
"N714" :	"FUEL PREBURNER AUGMENTED SPARK IGNITER OXIDIZER ORIFICE (016.1.3)"	0493
!		0494
"N715" :	"OXIDIZER PREBURNER OXIDIZER DOME PURGE ORIFICE (012.1.2)"	0495
!		0496
"N716" :	"FUEL PREBURNER OXIDIZER DOME PURGE ORIFICE (016.1.2)"	0497
!		0498
"N717" :	"MAIN COMBUSTION CHAMBER AUGMENTED SPARK IGNITER FUEL ORIFICE (F5.2)"	0499

Datatrieve Table FMEA_ITEM_NAME_TABLE (cont.)

!		0500
!	"N718" : "OXIDIZER PREBURNER AUGMENTED SPARK IGNITER FUEL ORIFICE (F25)"	0501
!		0502
!	"N719" : "FUEL PREBURNER AUGMENTED SPARK IGNITER FUEL ORIFICE (F21)"	0503
!		0504
!	"N720" :	0505
!	"MAIN COMBUSTION CHAMBER LIQUID O2 AUGMENTED SPARK IGNITER ORIFICE UPSTREAM"	0506
!		0507
!	"N721" :	0508
!	"MAIN COMBUSTION CHAMBER LIQUID O2 AUGMENTED SPARK IGNITER ORIFICE DOWNSTREAM"	0509
!		0510
!	"N722" :	0511
!	"MAIN COMBUSTION CHAMBER LIQUID O2 AUGMENTED SPARK IGNITER BYPASS ORIFICE"	0512
!		0513
!	"N723" : "HELIUM PRECHARGE VALVE DRAIN LINE ORIFICE"	0514
!		0515
!	"N724" : "GASEOUS OXYGEN CONTROL VALVE GASEOUS OXYGEN INLET ORIFICE (O24)"	0516
!		0517
!		0518
!	END_TABLE	0519

Datatrieve Table FMEA_ITEM_PART_NO_TABLE

DEFINE TABLE FMEA_ITEM_PART_NO_TABLE	0520
EDIT_STRING IS T(40)	0521
!	0522
!	0523
"A100" : "RS007051"	0524
!	0525
"A150" : "RS008801"	0526
!	0527
"A200" : "RS009122"	0528
!	0529
"A330" : "RS009170"	0530
!	0531
"A340" : "RS009168"	0532
!	0533
"A600" : "RS009020"	0534
!	0535
"A700" : "RS009004"	0536
!	0537
"B200" : "RS007501"	0538
!	0539
"B400" : "RS007701"	0540
!	0541
"B600" : "RS007601"	0542
!	0543
"B800" : "RS007801"	0544
!	0545
"C111" : "RS008059"	0546
!	0547
"C112" : "RS008059"	0548
!	0549
"C113" : "RS008059"	0550
!	0551
"C114" : "RS008059"	0552
!	0553
"C116" : "RS008059"	0554
!	0555
"C117" : "RS008059"	0556
!	0557
"C200" : "R0019450"	0558
!	0559
"C212" : "RS010341"	0560
!	0561
"C213" : "RS010341"	0562
!	0563
"C214" : "R0010725"	0564
!	0565
"C215" : "RS010341"	0566
!	0567
"C217" : "RS010341"	0568
!	0569
"C218" : "RS010341"	0570

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

!		0571
"C251" :	"RS008021"	0572
!		0573
"C252" :	"RS008021"	0574
!		0575
"C254" :	"RS008021"	0576
!		0577
"C255" :	"RS008021"	0578
!		0579
"C257" :	"R0010984"	0580
!		0581
"C258" :	"RS008021"	0582
!		0583
"C259" :	"RS019401"	0584
!		0585
"C260" :	"RS0011041"	0586
!		0587
"C300" :	"RS010180"	0588
!		0589
"D110" :	"RS008256"	0590
!		0591
"D120" :	"RS0082555"	0592
!		0593
"D130" :	"RS008257"	0594
!		0595
"D140" :	"RS008258"	0596
!		0597
"D150" :	"RS008259"	0598
!		0599
"D210" :	"RS008056"	0600
!		0601
"D220" :	"RS008056"	0602
!		0603
"D300" :	"R0019130"	0604
!		0605
"D500" :	"RS010141"	0606
!		0607
"D600" :	"RS010161"	0608
!		0609
"E110" :	"RES1008"	0610
!		0611
"E120" :	"RES1008"	0612
!		0613
"E130" :	"RES1008"	0614
!		0615
"E140" :	"RES1008"	0616
!		0617
"E150" :	"RES1008"	0618
!		0619
"F000" :	"RES1007"	0620
!		0621

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

"F800" : "RES1393"	0622
!	0623
"G100" : "RS003685"	0624
!	0625
"G200" : "RS003685"	0626
!	0627
"G300" : "RS003685"	0628
!	0629
"G400" : "RS003685"	0630
!	0631
"G500" : "RS003685"	0632
!	0633
"G600" : "RS003685"	0634
!	0635
"H101" : "RS008101"	0636
!	0637
"H102" : "RS008102"	0638
!	0639
"H103" : "RS008103"	0640
!	0641
"H104" : "RS008104"	0642
!	0643
"H105" : "RS008105"	0644
!	0645
"H107" : "RS008107"	0646
!	0647
"H108" : "RS008108"	0648
!	0649
"H109" : "RS008109"	0650
!	0651
"H110" : "RS008110"	0652
!	0653
"H111" : "RS008111"	0654
!	0655
"H112" : "RS008112"	0656
!	0657
"H113" : "RS008113"	0658
!	0659
"H114" : "RS008114"	0660
!	0661
"H115" : "RS008115"	0662
!	0663
"H116" : "RS008116"	0664
!	0665
"H118" : "RS008118"	0666
!	0667
"H119" : "RS008119"	0668
!	0669
"H120" : "RS008120"	0670
!	0671
"H122" : "RS008122"	0672

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

!		0673
"H128" :	"R0019511"	0674
!		0675
"J201" :	"RES7001"	0676
!		0677
"J202" :	"RES7001"	0678
!		0679
"J203" :	"RES7001"	0680
!		0681
"J205" :	"RES7001"	0682
!		0683
"J207" :	"RES7001"	0684
!		0685
"J208" :	"RES7001"	0686
!		0687
"J209" :	"RES7001"	0688
!		0689
"J210" :	"RES7001"	0690
!		0691
"J211" :	"RES7001"	0692
!		0693
"J213" :	"RES7001"	0694
!		0695
"J216" :	"RES7001"	0696
!		0697
"J218" :	"RES7001"	0698
!		0699
"J220" :	"RES7001"	0700
!		0701
"J221" :	"RES7001"	0702
!		0703
"J222" :	"RES7001"	0704
!		0705
"J223" :	"RES7001"	0706
!		0707
"J224" :	"RES7001"	0708
!		0709
"J225" :	"RES7001"	0710
!		0711
"J228" :	"RES7001"	0712
!		0713
"J229" :	"RES7001"	0714
!		0715
"J301" :	"RES7004"	0716
!		0717
"J302" :	"RES7004"	0718
!		0719
"J303" :	"RES7004"	0720
!		0721
"J304" :	"RES7004"	0722
!		0723

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

"J306" : "RES7002"	0724
!	0725
"J309" : "RES7002"	0726
!	0727
"J310" : "RES7002"	0728
!	0729
"J311" : "RES7002"	0730
!	0731
"J312" : "RES7002"	0732
!	0733
"J601" : "RES7005"	0734
!	0735
"J602" : "RES7005"	0736
!	0737
"J607" : "RES7005"	0738
!	0739
"J608" : "RES7005"	0740
!	0741
"J701" : "RS008241"	0742
!	0743
"K101" : "RS007018"	0744
!	0745
"K102" : "RS007034"	0746
!	0747
"K103" : "RS007037"	0748
!	0749
"K104" : "RS007043"	0750
!	0751
"K105" : "RS007012"	0752
!	0753
"K106" : "RS007026"	0754
!	0755
"K107" : "RS007046"	0756
!	0757
"K108" : "R0010748"	0758
!	0759
"K110" : "RS007168"	0760
!	0761
"K111" : "R0010758"	0762
!	0763
"K112" : "R0010751"	0764
!	0765
"K113" : "R0010752"	0766
!	0767
"K201" : "RS007015"	0768
!	0769
"K202" : "RS007035"	0770
!	0771
"K203" : "RES1221"	0772
!	0773
"K204" : "RS007016"	0774

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

!		0775
"K205" :	"RS007021"	0776
!		0777
"K206" :	"RS007031"	0778
!		0779
"K207" :	"RS007083"	0780
!		0781
"K208" :	"RS007029"	0782
!		0783
"K209" :	"RS007179"	0784
!		0785
"K210" :	"RS007228"	0786
!		0787
"K211" :	"RS007229"	0788
!		0789
"K212" :	"RS007032"	0790
!		0791
"K213" :	"RS007041"	0792
!		0793
"K214" :	"RS007297"	0794
!		0795
"K215" :	"RS007285"	0796
!		0797
"K216" :	"RS007369"	0798
!		0799
"K217" :	"RS007283"	0800
!		0801
"K218" :	"RS007284"	0802
!		0803
"K301" :	"RS007107, R0019438, R0019439"	0804
!		0805
"K302" :	"VARIOUS (7)"	0806
!		0807
"K303" :	"VARIOUS (6)"	0808
!		0809
"K304" :	"RS007118"	0810
!		0811
"K305" :	"RS007111"	0812
!		0813
"K306" :	"RS007163"	0814
!		0815
"K401" :	"RES1001"	0816
!		0817
"K402" :	"RS007212, RS007219, RS007220"	0818
!		0819
"K403" :	"RES1002"	0820
!		0821
"K501" :	"RES1004"	0822
!		0823
"K502" :	"RES1003"	0824
!		0825

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

"K504" : "RS007130"	0826
!	0827
"K505" : "RS007123"	0828
!	0829
"K506" : "RS007128"	0830
!	0831
"K507" : "RS007127"	0832
!	0833
"K508" : "RS007124"	0834
!	0835
"K509" : "RS007103"	0836
!	0837
"K510" : "RS007131"	0838
!	0839
"K511" : "RS007125, RS007126"	0840
!	0841
"K513" : "RS007171, R0019565"	0842
!	0843
"K514" : "R0019353"	0844
!	0845
"K515" : "RS007270"	0846
!	0847
"K516" : "RS007132"	0848
!	0849
"K517" : "RS007134"	0850
!	0851
"K518" : "RS007135"	0852
!	0853
"K519" : "RS007133"	0854
!	0855
"K520" : "RS007271"	0856
!	0857
"K530" : "RS007368"	0858
!	0859
"K531" : "RS007289"	0860
!	0861
"K532" : "RS007286"	0862
!	0863
"K533" : "RS007287"	0864
!	0865
"K534" : "RS007288"	0866
!	0867
"K600" : "RS010435"	0868
!	0869
"L101" : "VARIOUS"	0870
!	0871
"L102" : "VARIOUS"	0872
!	0873
"L103" : "VARIOUS"	0874
!	0875
"L104" : "VARIOUS"	0876

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

!		0877
"L105" :	"VARIOUS"	0878
!		0879
"M000" :	"RS008821"	0880
!		0881
"N300" :	"NOT APPLICABLE"	0882
!		0883
"N400" :	"RS007280"	0884
!		0885
"N701" :	"RS007159"	0886
!		0887
"N702" :	"RE251-4301"	0888
!		0889
"N703" :	"RS007184"	0890
!		0891
"N704" :	"RS007184"	0892
!		0893
"N705" :	"RS007184"	0894
!		0895
"N706" :	"RS007159"	0896
!		0897
"N707" :	"RS007159"	0898
!		0899
"N708" :	"RS007159"	0900
!		0901
"N709" :	"RS007352"	0902
!		0903
"N710" :	"RS008681"	0904
!		0905
"N711" :	"RS009038"	0906
!		0907
"N712" :	"RS009038"	0908
!		0909
"N713" :	"RS009100"	0910
!		0911
"N714" :	"RS009100"	0912
!		0913
"N715" :	"RS009100"	0914
!		0915
"N716" :	"RS009100"	0916
!		0917
"N717" :	"RS009546"	0918
!		0919
"N718" :	"RS009546"	0920
!		0921
"N719" :	"RS009546"	0922
!		0923
"N720" :	"RS009038"	0924
!		0925
"N721" :	"RS009038"	0926
!		0927

Datatrieve Table FMEA_ITEM_PART_NO_TABLE (cont.)

"N722" : "RS009038"	0928
!	0929
"N723" : "RE251-4301"	0930
!	0931
"N724" : "R0019972"	0932
!	0933
!	0934
END_TABLE	0935

Datatrieve Table FREQ_TIME_UNITS_TABLE

DEFINE TABLE FREQ_TIME_UNITS_TABLE	0936
!	0937
!	0938
"ACOUSTIC" : "HERTZ"	0939
!	0940
"ELECTRICAL" : "HERTZ"	0941
!	0942
"FLOW" : "HERTZ"	0943
!	0944
"PRESSURE" : "HERTZ"	0945
!	0946
"RPM" : "HERTZ"	0947
!	0948
"THERMAL" : "SECONDS"	0949
!	0950
"TORQUE" : "HERTZ"	0951
!	0952
"VIBRATION" : "HERTZ"	0953
!	0954
"WORN PARTICLES" : "SECONDS"	0955
!	0956
!	0957
END_TABLE	0958

Datatrieve Table MONTH_TABLE

DEFINE TABLE MONTH_TABLE	0959
!	0960
!	0961
"JAN" : "JANUARY"	0962
!	0963
"FEB" : "FEBRUARY"	0964
!	0965
"MAR" : "MARCH"	0966
!	0967
"APR" : "APRIL"	0968
!	0969
"MAY" : "MAY"	0970
!	0971
"JUN" : "JUNE"	0972
!	0973
"JUL" : "JULY"	0974
!	0975
"AUG" : "AUGUST"	0976
!	0977
"SEP" : "SEPTEMBER"	0978
!	0979
"OCT" : "OCTOBER"	0980
!	0981
"NOV" : "NOVEMBER"	0982
!	0983
"DEC" : "DECEMBER"	0984
!	0985
!	0986
END_TABLE	0987

Datatrieve Table NUMBER_TABLE

DEFINE TABLE NUMBER_TABLE

!

!

"0" : "0"

!

"1" : "1"

!

"2" : "2"

!

"3" : "3"

!

"4" : "4"

!

"5" : "5"

!

"6" : "6"

!

"7" : "7"

!

"8" : "8"

!

"9" : "9"

!

END_TABLE

0988

0989

0990

0991

0992

0993

0994

0995

0996

0997

0998

0999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

Datatrieve Table PARAMETER_TABLE

DEFINE TABLE PARAMETER_TABLE	1013
!	1014
!	1015
"AMPLITUDE" : "SAME AS SIGNAL UNITS"	1016
!	1017
"FREQUENCY" : "HERTZ"	1018
!	1019
"PHASE" : "DEGREES"	1020
!	1021
!	1022
END_TABLE	1023

Datatrieve Table REFERENCE_ABBREV_TABLE

DEFINE TABLE REFERENCE_ABBREV_TABLE	1024
!	1025
!	1026
"AEROJET" : "AJ"	1027
!	1028
"BATTELLE" : "BA"	1029
!	1030
"MARTIN MARIETTA" : "MM"	1031
!	1032
"NASA HDQ" : "NH"	1033
!	1034
"NASA MSFC" : "NM"	1035
!	1036
"PRATT & WHITNEY" : "PW"	1037
!	1038
"ROCKETDYNE" : "RD"	1039
!	1040
!	1041
END_TABLE	1042

Datatrieve Table REFERENCE_SOURCE_TABLE

DEFINE TABLE REFERENCE_SOURCE_TABLE	1043
!	1044
!	1045
"AEROJET" : "AEROJET TECHSYSTEMS COMPANY"	1046
!	1047
"BATTELLE" : "COLUMBUS DIVISION, BATTELLE MEMORIAL INSTITUTE"	1048
!	1049
"MARTIN MARIETTA" : "MARTIN MARIETTA DENVER AEROSPACE"	1050
!	1051
"NASA HDQ" : "NASA, HEADQUARTERS"	1052
!	1053
"NASA MSFC" : "NASA, GEORGE C. MARSHALL SPACE FLIGHT CENTER"	1054
!	1055
"PRATT & WHITNEY" :	1056
"GOVERNMENT PRODUCTS DIVISION, UNITED TECHNOLOGIES, PRATT & WHITNEY AIRCRAFT"	1057
!	1058
"ROCKETDYNE" : "ROCKETDYNE DIVISION, ROCKWELL INTERNATIONAL CORPORATION"	1059
!	1060
!	1061
END_TABLE	1062

Datatrieve Table SIGNAL_TABLE

DEFINE TABLE SIGNAL_TABLE	1063
!	1064
!	1065
"ACOUSTIC" : "ACOUSTIC EVENTS"	1066
!	1067
"ELECTRICAL" : "VOLTS"	1068
!	1069
"FLOW" : "LB-MASS PER SECOND"	1070
!	1071
"PRESSURE" : "PSIA"	1072
!	1073
"RPM" : "RPM"	1074
!	1075
"THERMAL" : "DEGREES-K"	1076
!	1077
"TORQUE" : "INCH-POUNDS"	1078
!	1079
"VIBRATION" : "ACCELERATION-G"	1080
!	1081
"WORN PARTICLES" : "PARTICLES PER SECOND"	1082
!	1083
!	1084
END_TABLE	1085

Datatrieve Table SIGN_TABLE

DEFINE TABLE SIGN_TABLE

1086

!
!

1087

"+" : "+"

1088

!

1089

"- " : "- "

1090

!

1091

!

1092

END_TABLE

1093

1094

G-32

(This page intentionally blank)

APPENDIX H

FIPM TDMS FORM DEFINITIONS

<u>Form Definition</u>	<u>Page</u>
CONNECTIONS_STO_FORM	H-3
FAILUREMODES_FIN1_FORM	H-7
FAILUREMODES_FIN2_FORM	H-13
FAILUREMODES_MOD1_FORM	H-19
FAILUREMODES_MOD2_FORM	H-25
FAILUREMODES_STO1_FORM	H-31
FAILUREMODES_STO2_FORM	H-37
MODULES_FIN_FORM	H-43
MODULES_MOD_FORM	H-47
MODULES_STO_FORM	H-53
PROPAGATIONS_FIN_FORM	H-57
PROPAGATIONS_MOD_FORM	H-63
PROPAGATIONS_STO_FORM	H-71
REFERENCES_FIN_FORM	H-79
REFERENCES_MOD_FORM	H-85
REFERENCES_STO_FORM	H-91
SYSTEMS_FIN_FORM	H-97
SYSTEMS_MOD_FORM	H-107
SYSTEMS_STO_FORM	H-117

H-2

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

CONNECTIONS_STO_FORM

Page 1

22-OCT-1987 14:17:41 VAX-11 FDU V1.4
22-OCT-1987 14:17:41 SYS\$INPUT:[] .COM; (1)

Form CONNECTIONS_STO_FORM
Form Definition

Form name: CONNECTIONS_STO_FORM
Form path name: _CDD\$TOP.FIPM.CONNECTIONS_STO_FORM
Help form name:
Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width: 80
Date/time form was stored in CDD: 17-NOV-1986 15:12:03.25

Field Access Order List:
Field name Subscript

SYSTEM_A
MODULE_A
CONNECTION
UNANTICIPATED
SYSTEM_B
MODULE_B
CONTINUE

ORIGINAL PAGE IS
OF POOR QUALITY

Page 2

CONNECTIONS_STO_FORM (CONTINUED)

22-OCT-1987 14:17:41 VAX-11 FDU V1.4
22-OCT-1987 14:17:41 SYS\$INPUT: [].COM; (1)

Form CONNECTIONS_STO_FORM
Form Definition

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
STORE CONNECTIONS							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
SYSTEM AND MODULE A: A999 9999							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
CONNECTION (TYPE AND QUALIFIER): AAXX							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
UNANTICIPATED CONNECTION (T OR F): A							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
SYSTEM AND MODULE B: A999 9999							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
CONTINUE (Y, N OR A): A							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
MESSAGE:							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
XX							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							

CONNECTIONS_STO_FORM (CONTINUED)

Form Definition CONNECTIONS_STO_FORM VAX-11 FDU V1.4 VAX-11 FDU V1.4
 Form Definition 22-OCT-1987 14:17:41 SYS\$INPUT: [].COM; (1) Page 3

FIELD DEFINITIONS

5,22 Field name: SYSTEM_A
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

5,27 Field name: MODULE_A
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

8,34 Field name: CONNECTION
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

11,36 Field name: UNANTICIPATED
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
 OF POOR QUALITY

CONNECTIONS_STO_FORM (CONTINUED)

Form Definition CONNECTIONS_STO_FORM 22-OCT-1987 14:17:41 VAX-11 FDU V1.4 Page 4
 Form Definition 22-OCT-1987 14:17:41 SYSINPUT: [].COM; (1)

14,22 Field name: SYSTEM_B
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

14,27 Field name: MODULE_B
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

17,51 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
OF EQUAL QUALITY

FAILUREMODES_FIN1_FORM

Form FAILUREMODES_FIN1_FORM Page 1
Form Definition VAX-11 FDU V1.4
22-OCT-1987 18:28:04
22-OCT-1987 18:28:04 SYS\$INPUT: [].COM; (1)

Form name: FAILUREMODES_FIN1_FORM
Form path name: _CDD\$TOP.FIPM.FAILUREMODES_FIN1_FORM
Help form name:
Help form path name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 5-JAN-1987 13:23:57.68

Field Access Order List:
Field name Subscript

SOURCE_SYSTEM
SOURCE_MODULE
MODE_SUBMODE
ACCOM_SYSTEM
ACCOM_MODULE
DESCRIPTION_1
DESCRIPTION_2
DESCRIPTION_3
EFFECT_1A
EFFECT_1B

FAILUREMODES_FINI_FORM (CONTINUED)

Form Definition FAILUREMODES_FINI_FORM Page 2
22-OCT-1987 16:26:04 VAX-11 FDU V1.4
22-OCT-1987 16:26:04 SYS\$INPUT: [].COM; (1)

FORM IMAGE

1	1	2	3	4	5	6	7	8
2	12345678901234567890123456789012345678901234567890							
3	FIND FAILURE MODES							
4	SOURCE SYSTEM AND MODULE: XXXX XXXX							
5	FAILURE MODE AND SUBMODE: XXXX							
6	ACCOMPLICE SYSTEM AND MODULE: XXXX XXXX							
7	FAILURE DESCRIPTION:							
8	XX							
9	XX							
10	XX							
11	XX							
12	XX							
13	XX							
14	XX							
15	FAILURE EFFECTS:							
16	1)XX							
17	XX							
18	XX							
19	XX							
20	XX							
21	MESSAGE:							
22	XX							
23	12345678901234567890123456789012345678901234567890							

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_FIN1_FORM (CONTINUED)

Form Definition FAILUREMODES_FIN1_FORM
22-OCT-1987 16:26:04 VAX-11 FDU V1.4
22-OCT-1987 16:26:04 SYS\$INPUT: [].COM; (1) Page 3

FIELD DEFINITIONS

4,27 Field name: SOURCE_SYSTEM
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

4,32 Field name: SOURCE_MODULE
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB,
Display attributes: REVERSE,

6,27 Field name: MODE_SUBMODE
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

8,31 Field name: ACCOM_SYSTEM
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

FAILUREMODES_FIN1_FORM (CONTINUED)

Page 4

22-OCT-1987 16:26:04 VAX-11 FDU V1.4
 22-OCT-1987 16:26:04 SYS\$INPUT: [].COM; (1)

Form FAILUREMODES_FIN1_FORM
 Form Definition

8,36
 Field name: ACCOM_MODULE
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB,
 Display attributes: REVERSE,

11,1
 Field name: DESCRIPTION_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

12,1
 Field name: DESCRIPTION_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

13,1
 Field name: DESCRIPTION_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
 OF POOR QUALITY

FAILUREMODES_FIN1_FORM (CONTINUED)

Form FAILUREMODES_FIN1_FORM
 Form Definition
 22-OCT-1987 16:26:04 VAX-11 FDU V1.4
 22-OCT-1987 16:26:04 SYS\$INPUT:[] .COM; (1)
 Page 5

17,1 Field name: EFFECT_1A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

18,1 Field name: EFFECT_1B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

23,1 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

H-12

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_FIN2_FORM

Page 1

22-OCT-1987 16:26:20 VAX-11 FDU V1.4
22-OCT-1987 16:26:20 SYSSINPUT: [] .COM; (1)

Form FAILUREMODES_FIN2_FORM
Form Definition

Form name: FAILUREMODES_FIN2_FORM
Form path name: _CDDSTOP.FIPM.FAILUREMODES_FIN2_FORM
Help form name:
Help form path name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 5-JAN-1987 13:26:33.26

Field Access Order List:
Field name Subscript

EFFECT_2A
EFFECT_2B
EFFECT_3A
EFFECT_3B
EFFECT_4A
EFFECT_4B
EFFECT_5A
EFFECT_5B
EFFECT_6A
EFFECT_6B
CONTINUE

FAILUREMODES_FIN2_FORM (CONTINUED)

Page 2

22-OCT-1987 16:26:20 VAX-11 FDU V1.4
22-OCT-1987 16:26:20 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_FIN2_FORM
Form Definition

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
2							
3							
4							
5							
6							

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_FIN2_FORM (CONTINUED)

Form Definition FAILUREMODES_FIN2_FORM
22-OCT-1987 16:26:20 VAX-11 FDU V1.4
22-OCT-1987 16:26:20 SYS\$INPUT: [] .COM; (1)
Page 3

FIELD DEFINITIONS

6,1 Field name: EFFECT_2A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,1 Field name: EFFECT_2B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

9,1 Field name: EFFECT_3A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

10,1 Field name: EFFECT_3B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

FAILUREMODES_FIN2_FORM (CONTINUED)

Page 4

22-OCT-1987 16:26:20 VAX-11 FDU V1.4
 22-OCT-1987 16:26:20 SYSSINPUT: [] .COM; (1)

Form FAILUREMODES_FIN2_FORM
 Form Definition

12,1 Field name: EFFECT_4A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

13,1 Field name: EFFECT_4B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

15,1 Field name: EFFECT_5A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

16,1 Field name: EFFECT_5B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_FIN2_FORM (CONTINUED)

Page 5

22-OCT-1987 16:26:20 VAX-11 FDU V1.4
22-OCT-1987 16:26:20 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_FIN2_FORM
Form Definition

18,1 Field name: EFFECT_6A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

19,1 Field name: EFFECT_6B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

21,50 Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

H-18

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_MOD1_FORM

Form FAILUREMODES_MOD1_FORM Page 1
Form Definition 22-OCT-1987 16:26:37 VAX-11 FDU V1.4
22-OCT-1987 16:26:37 SYS\$INPUT: [] .COM; (1)

Form name: FAILUREMODES_MOD1_FORM
Form path name: _CDD\$TOP.FIPM.FAILUREMODES_MOD1_FORM
Help form name:
Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width: 80
Date/time form was stored in CDD: 12-JAN-1987 16:53:04.91

Field Access Order List:
Field name Subscript

- DESCRIPTION_1
- DESCRIPTION_2
- DESCRIPTION_3
- EFFECT_1A
- EFFECT_1B

FAILUREMODES_MOD1_FORM (CONTINUED)

Form Definition FAILUREMODES_MOD1_FORM VAX-11 FDU V1.4 Page 2
22-OCT-1987 16:28:37 22-OCT-1987 16:28:37 SYSSINPUT:[].COM; (1)

FORM IMAGE

1	2	3	4	5	6	7	8
1	123456789012345678901234567890123456789012345678901234567890						
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							

123456789012345678901234567890123456789012345678901234567890
MODIFY FAILURE MODES
RECORD 99999 OF 99999
SOURCE SYSTEM AND MODULE: A999 9999
FAILURE MODE AND SUBMODE: AAAA
ACCOMPLICE SYSTEM AND MODULE: XXXX 9999
FAILURE DESCRIPTION:
FAILURE EFFECTS:
1)
MESSAGE:
123456789012345678901234567890123456789012345678901234567890

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_MOD1_FORM (CONTINUED)

Page 3

22-OCT-1987 16:26:37 VAX-11 FDU V1.4
22-OCT-1987 16:26:37 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_MOD1_FORM
Form Definition

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

5,27 Field name: SOURCE_SYSTEM
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ''
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

5,32 Field name: SOURCE_MODULE
Field length: 4
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

FAILUREMODES_MOD1_FORM (CONTINUED)

Page 4

22-OCT-1987 16:26:37 VAX-11 FDU V1.4
 22-OCT-1987 16:26:37 SYSSINPUT: [].COM; (1)

Form FAILUREMODES_MOD1_FORM
 Form Definition

7,27 Field name: MODE_SUBMODE
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

9,31 Field name: ACCOM_SYSTEM
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

9,36 Field name: ACCOM_MODULE
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: '0', , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

12,1 Field name: DESCRIPTION_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_MOD1_FORM (CONTINUED)

Form	FAILUREMODES_MOD1_FORM	22-OCT-1987 18:28:37	VAX-11 FDU V1.4	Page 5
Form Definition		22-OCT-1987 18:28:37	SYSSINPUT: [] .COM; (1)	

13,1 Field name: DESCRIPTION_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

14,1 Field name: DESCRIPTION_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

18,1 Field name: EFFECT_1A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

19,1 Field name: EFFECT_1B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_MOD1_FORM (CONTINUED)

Page 6

22-OCT-1987 16:26:37 VAX-11 FDU V1.4
22-OCT-1987 16:26:37 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_MOD1_FORM
Form Definition

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Display attributes: REVERSE, DISPLAY ONLY,

FAILUREMODES_MOD2_FORM

Form Definition
 Form name: FAILUREMODES_MOD2_FORM
 Form path name: _CDD\$TOP.FIPW.FAILUREMODES_MOD2_FORM
 Help form name: 1
 Beginning line number: 23
 Last line number: 80
 Form screen width: 80
 Date/time form was stored in CDD: 12-JAN-1987 15:54:35.75
 Page 1

Form name: FAILUREMODES_MOD2_FORM
 Form path name: _CDD\$TOP.FIPW.FAILUREMODES_MOD2_FORM
 Help form name: 1
 Beginning line number: 23
 Last line number: 80
 Form screen width: 80
 Date/time form was stored in CDD: 12-JAN-1987 15:54:35.75

Field Access Order List:
 Field name Subscript

EFFECT_2A
 EFFECT_2B
 EFFECT_3A
 EFFECT_3B
 EFFECT_4A
 EFFECT_4B
 EFFECT_5A
 EFFECT_5B
 EFFECT_6A
 EFFECT_6B
 CONTINUE

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_MOD2_FORM (CONTINUED)

Form Definition FAILUREMODES_MOD2_FORM
22-OCT-1987 16:28:49 VAX-11 FDU V1.4
22-OCT-1987 16:28:49 SYS\$INPUT:[] .COM; (1)
Page 3

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ','
Display attributes: REVERSE, DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ','
Display attributes: REVERSE, DISPLAY ONLY,

6,1 Field name: EFFECT_2A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ','
Clear character: ','
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,1 Field name: EFFECT_2B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ','
Clear character: ','
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

FAILUREMODES_MOD2_FORM (CONTINUED)

Page 4

22-OCT-1987 16:26:49 VAX-11 FDU V1.4
 22-OCT-1987 16:26:49 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_MOD2_FORM
 Form Definition

9,1 Field name: EFFECT_3A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

10,1 Field name: EFFECT_3B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

12,1 Field name: EFFECT_4A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

13,1 Field name: EFFECT_4B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_MOD2_FORM (CONTINUED)

Form	FAILUREMODES_MOD2_FORM	22-OCT-1987 16:28:49	VAX-11 FDU V1.4	Page 5
Form Definition		22-OCT-1987 16:28:49	SYSSINPUT: [].COM; (1)	

15,1 Field name: EFFECT_5A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

16,1 Field name: EFFECT_5B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

18,1 Field name: EFFECT_5A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

19,1 Field name: EFFECT_5B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_MOD2_FORM (CONTINUED)

Form FAILUREMODES_MOD2_FORM
 Form Definition
 22-OCT-1987 16:28:49 VAX-11 FDU V1.4
 22-OCT-1987 16:28:49 SYS\$INPUT: [] .COM; (1)
 Page 6

21,51
 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1
 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_ST01_FORM

Page 1

VAX-11 FDU V1.4
SYS\$INPUT: [].COM; (1)

22-OCT-1987 15:36:00
22-OCT-1987 15:36:00

Form FAILUREMODES_ST01_FORM
Form Definition

Form name: FAILUREMODES_ST01_FORM
Form path name: _CDD\$TOP.FIPM.FAILUREMODES_ST01_FORM
Help form name:
Help form path name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 13-NOV-1986 09:47:14.99

Field Access Order List:
Field name Subscript

SOURCE_SYSTEM
SOURCE_MODULE
MODE_SUBMODE
ACCOM_SYSTEM
ACCOM_MODULE
DESCRIPTION_1
DESCRIPTION_2
DESCRIPTION_3
EFFECT_1A
EFFECT_1B

FAILUREMODES_ST01_FORM (CONTINUED)

Form	Form Definition	Form	Form Definition
FAILUREMODES_ST01_FORM		VAX-11 FDU V1.4	
		22-OCT-1987 15:36:00	
		22-OCT-1987 15:36:00	SYS\$INPUT: [] .COM; (1)

FORM IMAGE

	1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890								
	STORE FAILURE MODES							
1								
2								
3								
4	SOURCE SYSTEM AND MODULE:	A999	9999					
5								
6	FAILURE MODE AND SUBMODE:	AAAA						
7								
8	ACCOMPlice SYSTEM AND MODULE:	XXXX	9999					
9								
10	FAILURE DESCRIPTION:							
11	XX							
12	XX							
13	XX							
14	XX							
15	FAILURE EFFECTS:							
16	1)							
17	XX							
18	XX							
19	XX							
20								
21								
22	MESSAGE:							
23	XX							
123456789012345678901234567890123456789012345678901234567890								
	1	2	3	4	5	6	7	8

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_ST01_FORM (CONTINUED)

Form FAILUREMODES_ST01_FORM 22-OCT-1987 15:36:00 VAX-11 FDU V1.4 Page 3
 Form Definition 22-OCT-1987 15:36:00 SYS\$INPUT:[] .COM; (1)

FIELD DEFINITIONS

4,27 Field name: SOURCE_SYSTEM
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

4,32 Field name: SOURCE_MODULE
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

6,27 Field name: MODE_SUBMODE
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

8,31 Field name: ACCOM_SYSTEM
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_ST01_FORM (CONTINUED)

Page 4

22-OCT-1987 15:36:00 VAX-11 FDU V1.4
 22-OCT-1987 15:36:00 SYS\$INPUT: [].COM; (1)

Form FAILUREMODES_ST01_FORM
 Form Definition

8,36 Field name: ACCOM_MODULE
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

11,1 Field name: DESCRIPTION_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

12,1 Field name: DESCRIPTION_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

13,1 Field name: DESCRIPTION_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
 OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_ST01_FORM (CONTINUED)

Page 5

22-OCT-1987 15:36:00 VAX-11 FDU V1.4
22-OCT-1987 15:36:00 SYSSINPUT: [] .COM; (1)

Form FAILUREMODES_ST01_FORM
Form Definition

17,1 Field name: EFFECT_1A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

18,1 Field name: EFFECT_1B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

H-36

(This page intentionally blank)

FAILUREMODES_ST02_FORM

Page 1

VAX-11 FDU VI.4
SYSINPUT: [].COM; (1)

22-OCT-1987 16:27:08
22-OCT-1987 16:27:08

FAILUREMODES_ST02_FORM

Form Definition

Form name: FAILUREMODES_ST02_FORM
Form path name: _CDD\$TOP.FIPW.FAILUREMODES_ST02_FORM

Help form name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 13-NOV-1986 09:47:28.69

Field Access Order List:
Field name Subscript

EFFECT_2A
EFFECT_2B
EFFECT_3A
EFFECT_3B
EFFECT_4A
EFFECT_4B
EFFECT_5A
EFFECT_5B
EFFECT_6A
EFFECT_6B
CONTINUE

FAILUREMODES_ST02_FORM (CONTINUED)

Form Definition FAILUREMODES_ST02_FORM VAX-11 FDU V1.4 Page 2
22-OCT-1987 16:27:08 SYS\$INPUT: [].COM; (1)

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
12345678901234567890123456789012345678901234567890							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							

FAILUREMODES_ST02_FORM (CONTINUED)

Form Definition FAILUREMODES_ST02_FORM
 22-OCT-1987 16:27:08 VAX-11 FDU V1.4
 22-OCT-1987 16:27:08 SYS\$INPUT: [] .COM; (1)
 Page 3

FIELD DEFINITIONS

8,1 Field name: EFFECT_2A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

7,1 Field name: EFFECT_2B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

9,1 Field name: EFFECT_3A
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

10,1 Field name: EFFECT_3B
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

FAILUREMODES_ST02_FORM (CONTINUED)

Page 4

22-OCT-1987 16:27:08 VAX-11 FDU V1.4
22-OCT-1987 16:27:08 SYS\$INPUT: [] .COM; (1)

Form FAILUREMODES_ST02_FORM
Form Definition

12,1 Field name: EFFECT_4A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

13,1 Field name: EFFECT_4B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

15,1 Field name: EFFECT_5A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

16,1 Field name: EFFECT_5B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

H-40

ORIGINAL PAGE IS
OF POOR QUALITY

FAILUREMODES_ST02_FORM (CONTINUED)

Form Definition FAILUREMODES_ST02_FORM

22-OCT-1987 16:27:08 VAX-11 FDU V1.4 Page 5
22-OCT-1987 16:27:08 SYS\$INPUT: []:COM, (1)

18,1 Field name: EFFECT_6A
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

19,1 Field name: EFFECT_6B
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

21,51 Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

H-42

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

MODULES_FIN_FORM

Page 1

22-OCT-1987 18:30:14 VAX-11 FDU V1.4
22-OCT-1987 18:30:14 SYS\$INPUT: [].COM; (1)

Form MODULES_FIN_FORM
Form Definition

Form name: MODULES_FIN_FORM
Form path name: _CDD\$TOP.FIPM.MODULES_FIN_FORM

Help form name:
Help form path name: 1

Beginning line number: 23
Last line number: 80

Form screen width:
Date/time form was stored in CDD: 12-JAN-1987 09:43:34.36

Field Access Order List:
Field name Subscript

SYSTEM
MODULE
MODULE_NAME
FUNCTION_1
FUNCTION_2
FUNCTION_3
CONTINUE

MODULES_FIN_FORM (CONTINUED)

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
FIND MODULES							
1	2	3	4	5	6	7	8
SYSTEM: XXXX							
MODULE: XXXX							
MODULE NAME:							
XX							
MODULE FUNCTION:							
XX							
XX							
XX							
XX							
CONTINUE (Y OR N): A							
MESSAGE:							
XX							
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8

ORIGINAL PAGE IS
OF POOR QUALITY

MODULES_FIN_FORM (CONTINUED)

Page 3

22-OCT-1987 16:30:14 VAX-11 FDU V1.4
22-OCT-1987 16:30:14 SYS\$INPUT: [].COM; (1)

Form Definition MODULES_FIN_FORM

FIELD DEFINITIONS

5,9 Field name: SYSTEM
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

7,9 Field name: MODULE
Field length: 4
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB,
Display attributes: REVERSE,

10,1 Field name: MODULE_NAME
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

13,1 Field name: FUNCTION_1
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

MODULES_FIN_FORM (CONTINUED)

Page 4

22-OCT-1987 16:30:14 VAX-11 FDU V1.4
22-OCT-1987 16:30:14 SYS\$INPUT: [] .COM; (1)

Form MODULES_FIN_FORM
Form Definition

14,1 Field name: FUNCTION_2
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

15,1 Field name: FUNCTION_3
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

18,50 Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
OF POOR QUALITY

MODULES_MOD_FORM

Form name: MODULES_MOD_FORM
Form Definition
22-OCT-1987 16:30:27 VAX-11 FDU V1.4
22-OCT-1987 16:30:27 SYSSINPUT: [].COM; (1)
Page 1

Form name: MODULES_MOD_FORM
Form path name: _CDD\$TOP.FIPM.MODULES_MOD_FORM
Help form name:
Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width:
Date/time form was stored in CDD: 12-JAN-1987 17:46:44.69

Field Access Order List:
Field name Subscript

MODULE_NAME
FUNCTION_1
FUNCTION_2
FUNCTION_3
CONTINUE

MODULES_MOD_FORM (CONTINUED)

22-OCT-1987 16:30:27 VAX-11 FDU V1.4
22-OCT-1987 16:30:27 SYS\$INPUT: [].COM; (1)

Form Definition MODULES_MOD_FORM

FORM IMAGE

1	123456789012345678901234567890123456789012345678901234567890	1	8					
2		2						
3		3						
4		4						
5		5						
6	SYSTEM: A999	6						
7		7						
8	MODULE: 9999	8						
9		9						
10	MODULE NAME:	10						
11	XX	11						
12		12						
13	MODULE FUNCTION:	13						
14	XX	14						
15	XX	15						
16	XX	16						
17		17						
18		18						
19		19						
20		20						
21		21						
22	MESSAGE:	22						
23	XX	23						
	123456789012345678901234567890123456789012345678901234567890							
	1	2	3	4	5	6	7	8

MODULES_MOD_FORM (CONTINUED)

Form MODULES_MOD_FORM
Form Definition
22-OCT-1987 16:30:27 VAX-11 FDU V1.4
22-OCT-1987 16:30:27 SYS\$INPUT: [] .COM; (1)
Page 3

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

6,9 Field name: SYSTEM
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ''
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

8,9 Field name: MODULE
Field length: 4
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

MODULES_MOD_FORM (CONTINUED)

Form Definition MODULES_MOD_FORM VAX-11 FDU V1.4 Page 4
 22-OCT-1987 16:30:27 22-OCT-1987 16:30:27 SYS\$INPUT:[] .COM, (1)

11,1 Field name: MODULE_NAME
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

14,1 Field name: FUNCTION_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

15,1 Field name: FUNCTION_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

16,1 Field name: FUNCTION_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

MODULES_MOD_FORM (CONTINUED)

Form MODULES_MOD_FORM
 Form Definition 22-OCT-1987 16:30:27 VAX-11 FDU V1.4
 22-OCT-1987 16:30:27 SYSINPUT: [].COM; (1) Page 5

19,51
 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1
 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

H-52

(This page intentionally blank)

MODULES_STO_FORM

Form Definition MODULES_STO_FORM 22-OCT-1987 18:30:40 VAX-11 FDU V1.4 Page 1
 22-OCT-1987 18:30:40 SYS\$INPUT: [] : CDM; (1)

Form name: MODULES_STO_FORM
 Form path name: _CDD\$TOP.FIPM.MODULES_STO_FORM
 Help form name:
 Help form path name: 1
 Beginning line number: 23
 Last line number: 80
 Form screen width:
 Date/time form was stored in CDD: 13-NOV-1986 09:46:59.10

Field Access Order List:
 Field name Subscript

SYSTEM
 MODULE
 MODULE_NAME
 FUNCTION_1
 FUNCTION_2
 FUNCTION_3
 CONTINUE

FORM IMAGE

1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890							
SYSTEM: A999							
MODULE: 9999							
MODULE NAME:							
XX							
MODULE FUNCTION:							
XX							
XX							
XX							
XX							
MESSAGE:							
XX							
123456789012345678901234567890123456789012345678901234567890							

MODULES_STO_FORM (CONTINUED)

Form Definition MODULES_STO_FORM
 22-OCT-1987 16:30:40 VAX-11 FDU V1.4
 22-OCT-1987 16:30:40 SYS\$INPUT: [] .COM; (1) Page 3

FIELD DEFINITIONS

5,9 Field name: SYSTEM
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

7,9 Field name: MODULE
 Field length: 4
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

10,1 Field name: MODULE_NAME
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

13,1 Field name: FUNCTION_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

MODULES_STO_FORM (CONTINUED)

Form Definition MODULES_STO_FORM
 22-OCT-1987 16:30:40 VAX-11 FDU V1.4
 22-OCT-1987 16:30:40 SYS\$INPUT: []:COM; (1)

Page 4

14,1 Field name: FUNCTION_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

15,1 Field name: FUNCTION_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

16,51 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

H-57

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_FIN_FORM

Form Definition
PROPAGATIONS_FIN_FORM
22-OCT-1987 16:28:02 VAX-11 FDU V1.4
22-OCT-1987 16:28:02 SYS\$INPUT:[] .COM; (1)
Page 1

Form name: PROPAGATIONS_FIN_FORM
Form path name: _CDD\$TOP.FIPM.PROPAGATIONS_FIN_FORM
Help form name:
Help form path name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 12-JAN-1987 10:09:44.80

Field Access Order List:
Field name Subscript

CODE_NUMBER
QUALITY
FMCODE
SIGNAL_TYPE
DIMENSIONS
MAX_FREQ_TIME
MIN_FREQ_TIME
PARAM
DURATION
ONSET
FAILURE
COMMENT_1
COMMENT_2
COMMENT_3
CONTINUE

PROPAGATIONS_FIN_FORM (CONTINUED)

Page 2

22-OCT-1987 16:28:02 VAX-11 FDU V1.4
22-OCT-1987 16:28:02 SYS\$INPUT:[] .COM; (1)

Form Definition
PROPAGATIONS_FIN_FORM

FORM IMAGE

1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890							
FIND FAILURE INFORMATION PROPAGATIONS							
1							
2							
3							
4	CODE NUMBER: XXXXXXXXXXXXXXXXXX SIGNAL QUALITY: X						
5							
6	FM CODE: XXXXXXXXXXXXXXXXXX SIGNAL TYPE: XXXXXXXXXXXXXXXXXX DIMENSIONS: X						
7							
8	MAX. FREQ./TIME: XXX MIN. FREQ./TIME: XXX PARAMETER: XXXXXXXXXXXXXXXXXX						
9							
10	SYMPTOM DURATION: XXX PERIOD OF ONSET: XXX INDICATES FAILURE: X						
11							
12	COMMENTS:						
13	1) XXXXXXXXXXXXXXXXXX						
14	2) XXXXXXXXXXXXXXXXXX						
15	3) XXXXXXXXXXXXXXXXXX						
16	XX						
17	XX						
18	XX						
19	CONTINUE (Y OR N): A						
20							
21							
22	MESSAGE:						
23	XX						
123456789012345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8

ORIGINAL PAGE IS
OF POOR QUALITY

Page 3

PROPAGATIONS_FIN_FORM (CONTINUED)

22-OCT-1987 16:28:02 VAX-11 FDU V1.4
22-OCT-1987 16:28:02 SYS\$INPUT: [].COM; (1)

Form PROPAGATIONS_FIN_FORM
Form Definition

FIELD DEFINITIONS

4,14 Field name: CODE_NUMBER
Field length: 21
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: BOLD, REVERSE,

4,54 Field name: QUALITY
Field length: 1
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB,
Display attributes: REVERSE,

6,9 Field name: FMCODE
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: BOLD, REVERSE,

6,45 Field name: SIGNAL_TYPE
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: BOLD, REVERSE,

PROPAGATIONS_FIN_FORM (CONTINUED)

Page 4

22-OCT-1987 16:28:02 VAX-11 FDU VI.4
 22-OCT-1987 16:28:02 SYS\$INPUT: [].COM; (1)

Form PROPAGATIONS_FIN_FORM
 Form Definition

8,80 Field name: DIMENSIONS
 Field length: 1
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB,
 Display attributes: REVERSE,

8,18 Field name: MAX_FREQ_TIME
 Field length: 3
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB,
 Display attributes: REVERSE,

8,41 Field name: MIN_FREQ_TIME
 Field length: 3
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB,
 Display attributes: REVERSE,

8,58 Field name: PARAM
 Field length: 20
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_FIN_FORM (CONTINUED)

Form Definition PROPAGATIONS_FIN_FORM VAX-11 FDU V1.4 Page 5
22-OCT-1987 16:28:02 22-OCT-1987 16:28:02 SYSINPUT: [] .COM; (1)

10,19 Field name: DURATION
Field length: 3
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB,
Display attributes: REVERSE,

10,42 Field name: ONSET
Field length: 3
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB,
Display attributes: REVERSE,

10,67 Field name: FAILURE
Field length: 1
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

14,1 Field name: COMMENT_1
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

PROPAGATIONS_FIN_FORM (CONTINUED)

Page 6

22-OCT-1987 16:28:02 VAX-11 FDU V1.4
 22-OCT-1987 16:28:02 SYS\$INPUT: [].COM; (1)

Form PROPAGATIONS_FIN_FORM
 Form Definition

16,1 Field name: COMMENT_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

16,1 Field name: COMMENT_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

20,50 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_MOD_FORM

Form Definition
 PROPAGATIONS_MOD_FORM
 22-OCT-1987 16:28:20 VAX-11 FDU V1.4
 22-OCT-1987 16:28:20 SYS\$INPUT: [] :CDM; (1)
 Page 1

Form name: PROPAGATIONS_MOD_FORM
 Form path name: _CDD\$TOP.FIPM.PROPAGATIONS_MOD_FORM
 Help form name:
 Help form path name:
 Beginning line number: 1
 Last line number: 23
 Form screen width: 80
 Date/time form was stored in CDD: 12-JAN-1987 15:58:25.03

Field Access Order List:
 Field name Subscript

QUALITY
 DIMENSIONS
 MAX_FREQ_TIME
 MIN_FREQ_TIME
 DURATION
 ONSET
 FAILURE
 COMMENT_1
 COMMENT_2
 COMMENT_3
 CONTINUE

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_MOD_FORM (CONTINUED)

22-OCT-1987 16:28:20 VAX-11 FDU V1.4
22-OCT-1987 16:28:20 SYS\$INPUT: [] .COM; (1)

Form Definition PROPAGATIONS_MOD_FORM

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ''
Display attributes: DISPLAY ONLY,

5,14 Field name: CODE_NUMBER
Field length: 21
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ''
Clear character: ''
Display attributes: REVERSE, DISPLAY ONLY,

5,54 Field name: QUALITY
Field length: 1
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: ''
Clear character: ''
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

PROPAGATIONS_MOD_FORM (CONTINUED)

Form Definition PROPAGATIONS_MOD_FORM 22-OCT-1987 16:28:20 VAX-11 FDU V1.4 Page 4
 22-OCT-1987 16:28:20 SYS\$INPUT: [] .COM; (1)

7,9 Field name: FMCODE
 Field length: 20
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

7,45 Field name: SIGNAL_TYPE
 Field length: 20
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

7,80 Field name: DIMENSIONS
 Field length: 1
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

9,18 Field name: MAX_FREQ_TIME
 Field length: 3
 Field scale factor: 0
 Field picture type: SIGNED NUMERIC
 Field datatype: SIGNED NUMERIC
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_MOD_FORM (CONTINUED)

Form Definition PROPAGATIONS_MOD_FORM VAX-11 FDU V1.4 Page 5
22-OCT-1987 16:28:20 SYSSINPUT: [] .COM; (1)
22-OCT-1987 16:28:20

9,41 Field name: MIN_FREQ_TIME
Field length: 3
Field scale factor: 0
Field picture type: SIGNED NUMERIC
Field datatype: SIGNED NUMERIC
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

9,58 Field name: PARAM
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Display attributes: REVERSE, DISPLAY ONLY,

11,19 Field name: DURATION
Field length: 3
Field scale factor: 0
Field picture type: SIGNED NUMERIC
Field datatype: SIGNED NUMERIC
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

11,42 Field name: ONSET
Field length: 3
Field scale factor: 0
Field picture type: SIGNED NUMERIC
Field datatype: SIGNED NUMERIC
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

PROPAGATIONS_MOD_FORM (CONTINUED)

Page 6

22-OCT-1987 16:28:20 VAX-11 FDU VI.4
 22-OCT-1987 16:28:20 SYSINPUT: [].COM; (1)

Form PROPAGATIONS_MOD_FORM
 Form Definition

11,87 Field name: FAILURE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

15,1 Field name: COMMENT_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

17,1 Field name: COMMENT_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

19,1 Field name: COMMENT_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

PROPAGATIONS_MOD_FORM (CONTINUED)

Form	PROPAGATIONS_MOD_FORM	22-OCT-1987 16:28:20	VAX-11 FDU V1.4	Page 7
Form Definition		22-OCT-1987 16:28:20	SYS\$INPUT: [].COM; (1)	

21,51

Field name: CONTINUE

Field length: 1

Field picture type: ALPHABETIC

Field datatype: TEXT

Fill character: , ,

Clear character: , ,

Attributes assigned: RESPONSE REQUIRED, UPPERCASE,

Display attributes: REVERSE,

23,1

Field name: MESSAGE

Field length: 80

Field picture type: ALPHANUMERIC

Field datatype: TEXT

Fill character: , ,

Clear character: , ,

Display attributes: REVERSE, DISPLAY ONLY,

H-70

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

Page 1

PROPAGATIONS_STO_FORM

VAX-11 FDU V1.4
SYSSINPUT: [].COM; (1)

22-OCT-1987 16:28:33
22-OCT-1987 16:28:33

Form Definition
PROPAGATIONS_STO_FORM

Form name: PROPAGATIONS_STO_FORM
Form path name: _CDD\$TOP.FIPM.PROPAGATIONS_STO_FORM
Help form name:
Help form path name:
Beginning line number: 1
Last line number: 23
Form screen width: 80
Date/time form was stored in CDD: 5-JAN-1987 13:10:29.17

Field Access Order List:
Field name Subscript

CODE_NUMBER
QUALITY
FMCODE
SIGNAL_TYPE
DIMENSIONS
MAX_FREQ_TIME
MIN_FREQ_TIME
PARAM
DURATION
ONSET
FAILURE
COMMENT_1
COMMENT_2
COMMENT_3
CONTINUE
FAILSAFE

PRECEDING PAGE BLANK NOT FILMED

PROPAGATIONS_STO_FORM (CONTINUED)

Form	PROPAGATIONS_STO_FORM	22-OCT-1987 16:28:33	VAX-11 FDU V1.4	Page 2
Form		22-OCT-1987 16:28:33	SYS\$INPUT:[] .COM; (1)	

FORM IMAGE

[illegible]

PROPAGATIONS_STO_FORM (CONTINUED)

Form Definition PROPAGATIONS_STO_FORM 22-OCT-1987 16:28:33 VAX-11 FDU V1.4 Page 3
 Form Definition 22-OCT-1987 16:28:33 SYS\$INPUT:[]:COM; (1)

FIELD DEFINITIONS

4,14 Field name: CODE_NUMBER
 Field length: 21
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

4,54 Field name: QUALITY
 Field length: 1
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

6,9 Field name: FWCODE
 Field length: 20
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

6,45 Field name: SIGNAL_TYPE
 Field length: 20
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ', '
 Clear character: ', '
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

PROPAGATIONS_STO_FORM (CONTINUED)

Form Definition PROPAGATIONS_STO_FORM 22-OCT-1987 16:28:33 VAX-11 FDU V1.4 Page 4
 22-OCT-1987 16:28:33 SYS\$INPUT: [] .COM; (1)

8,80 Field name: DIMENSIONS
 Field length: 1
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

8,18 Field name: MAX_FREQ_TIME
 Field length: 3
 Field scale factor: 0
 Field picture type: SIGNED NUMERIC
 Field datatype: SIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

8,41 Field name: MIN_FREQ_TIME
 Field length: 3
 Field scale factor: 0
 Field picture type: SIGNED NUMERIC
 Field datatype: SIGNED NUMERIC
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
 Display attributes: REVERSE,

8,58 Field name: PARAM
 Field length: 20
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_STO_FORM (CONTINUED)

Form Definition PROPAGATIONS_STO_FORM
22-OCT-1987 16:28:33 VAX-11 FDU V1.4
22-OCT-1987 16:28:33 SYSSINPUT: [].COM; (1) Page 5

10,19 Field name: DURATION
Field length: 3
Field scale factor: 0
Field picture type: SIGNED NUMERIC
Field datatype: SIGNED NUMERIC
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

10,42 Field name: ONSET
Field length: 3
Field scale factor: 0
Field picture type: SIGNED NUMERIC
Field datatype: SIGNED NUMERIC
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED,
Display attributes: REVERSE,

10,67 Field name: FAILURE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

14,1 Field name: COMMENT_1
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

PROPAGATIONS_STO_FORM (CONTINUED)

Page 6

22-OCT-1987 16:28:33 VAX-11 FDU V1.4
 22-OCT-1987 16:28:33 SYS\$INPUT: [] .COM; (1)

Form PROPAGATIONS_STO_FORM
 Form Definition

16,1 Field name: COMMENT_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

18,1 Field name: COMMENT_3
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

20,24 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

20,79 Field name: FAILSAFE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

H-77

ORIGINAL PAGE IS
OF POOR QUALITY

PROPAGATIONS_STO_FORM (CONTINUED)

Form	PROPAGATIONS_STO_FORM	22-OCT-1987 16:28:33	VAX-11 FDU V1.4	Page 7
Form Definition		22-OCT-1987 16:28:33	SYS\$INPUT: [] .COM; (1)	
23,1	Field name: MESSAGE			
	Field length: 80			
	Field picture type: ALPHANUMERIC			
	Field datatype: TEXT			
	Fill character: ,			
	Clear character: ,			
	Display attributes: REVERSE, DISPLAY ONLY,			

H-78

(This page intentionally blank)

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_FIN_FORM

Form REFERENCES_FIN_FORM
Form Definition

22-OCT-1987 16:29:11 VAX-11 FDU V1.4
22-OCT-1987 16:29:11 SYS\$INPUT: [] .COM; (1)

Form name: REFERENCES_FIN_FORM
Form path name: _CDD\$TOP.FIPM.REFERENCES_FIN_FORM
Help form name:

Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width: 80
Date/time form was stored in CDD: 20-JAN-1987 20:11:25.71

Field Access Order List:

Field name Subscript

AUTHOR_1
AUTHOR_2
AUTHOR_3
AUTHOR_4
TITLE_1
TITLE_2
ORGANIZATION
DOCUMENT_NO
DATE
CONTRACT_NO
CONTINUE

VAX-11 FDU V1.4
SYS\$INPUT:[].COM; (1)

Form	Form Definition	REFERENCES_FIN_FORM

FORM IMAGE

	1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890								
FIND REFERENCES								
AUTHORS (LAST NAME, INITIALS):								
1) XXXXXXXXXXXXXXXXXXXXXXXX	3)	XXXXXXXXXXXXXXXXXXXXXXXXXX						
2) XXXXXXXXXXXXXXXXXXXXXXXX	4)	XXXXXXXXXXXXXXXXXXXXXXXXXX						
REPORT OR DOCUMENT TITLE:								
ORIGINATING ORGANIZATION: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX								
ORIGINATING ORGANIZATION'S DOCUMENT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXXXX								
DOCUMENT DATE (DD-MMM-YYYY): XX-XXX-XXXX								
CONTRACT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXXXX								
CONTINUE (Y OR N): A								
MESSAGE:								
123456789012345678901234567890123456789012345678901234567890								

REFERENCES_FIN_FORM (CONTINUED)

Form REFERENCES_FIN_FORM
Form Definition

22-OCT-1987 16:29:11 VAX-11 FDU V1.4
22-OCT-1987 16:29:11 SYS\$INPUT: [].COM; (1)

Page 3

FIELD DEFINITIONS

6,7 Field name: AUTHOR_1
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

6,43 Field name: AUTHOR_3
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,7 Field name: AUTHOR_2
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,43 Field name: AUTHOR_4
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

H-81

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_FIN_FORM (CONTINUED)

Form Definition REFERENCES_FIN_FORM
 22-OCT-1987 16:29:11 VAX-11 FDU V1.4
 22-OCT-1987 16:29:11 SYS\$INPUT: [] .COM; (1) Page 4

10,1
 Field name: TITLE_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPCASE,
 Display attributes: REVERSE,

11,1
 Field name: TITLE_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPCASE,
 Display attributes: REVERSE,

13,27
 Field name: ORGANIZATION
 Field length: 30
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPCASE,
 Display attributes: REVERSE,

15,45
 Field name: DOCUMENT_NO
 Field length: 30
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPCASE,
 Display attributes: REVERSE,

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_FIN_FORM (CONTINUED)

Form REFERENCES_FIN_FORM Page 5
Form Definition 22-OCT-1987 18:29:11 VAX-11 FDU V1.4
22-OCT-1987 18:29:11 SYS\$INPUT: [].COM; (1)

17,30
Field name: DATE
Field length: 9
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, UPPERCASE,
Display attributes: REVERSE,

19,18
Field name: CONTRACT_NO
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

21,50
Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

23,1
Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Display attributes: REVERSE, DISPLAY ONLY,

H-84

(This page intentionally blank)

H-85

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_MOD_FORM

Form REFERENCES_MOD_FORM Page 1
Form Definition 22-OCT-1987 16:29:37 VAX-11 FDU V1.4
22-OCT-1987 16:29:37 SYS\$INPUT:[]:COM; (1)

Form name: REFERENCES_MOD_FORM
Form path name: _CDD\$TOP.FIPM.REFERENCES_MOD_FORM
Help form name:
Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width: 80
Date/time form was stored in CDD: 21-JAN-1987 15:22:59.00

Field Access Order List:
Field name Subscript

AUTHOR_1
AUTHOR_2
AUTHOR_3
AUTHOR_4
DOCUMENT_NO
CONTRACT_NO
CONTINUE

PRELIMINARY PAGE BEING NOT FILMED

REFERENCES_MOD_FORM (CONTINUED)

Page 2

22-OCT-1987 16:29:37 VAX-11 FDU V1.4
 22-OCT-1987 16:29:37 SYS\$INPUT: [] .COM; (1)

Form REFERENCES_MOD_FORM
 Form Definition

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
MODIFY REFERENCES							
RECORD 99999 OF 99999							
AUTHORS (LAST NAME, INITIALS):							
1) XXXXXXXXXXXXXXXXXXXXXXXX 3) XXXXXXXXXXXXXXXXXXXXXXXX							
2) XXXXXXXXXXXXXXXXXXXXXXXX 4) XXXXXXXXXXXXXXXXXXXXXXXX							
REPORT OR DOCUMENT TITLE:							
XX							
XX							
XX							
ORIGINATING ORGANIZATION: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX							
ORIGINATING ORGANIZATION'S DOCUMENT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXX							
DOCUMENT DATE (DD-MMM-YYYY): 99XXXXXX9999							
CONTRACT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXX							
CONTINUE (Y, N OR A): A							
MESSAGE:							
XX							
12345678901234567890123456789012345678901234567890							

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_MOD_FORM (CONTINUED)

Page 3

22-OCT-1987 16:29:37 VAX-11 FDU V1.4
22-OCT-1987 16:29:37 SYS\$INPUT: [] .COM; (1)

Form REFERENCES_MOD_FORM
Form Definition

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ','
Display attributes: REVERSE, DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
Field length: 5
Field scale factor: 0
Field picture type: UNSIGNED NUMERIC
Field datatype: UNSIGNED NUMERIC
Fill character: '0'
Clear character: ','
Display attributes: REVERSE, DISPLAY ONLY,

6,7 Field name: AUTHOR_1
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ','
Clear character: ','
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

6,43 Field name: AUTHOR_3
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ','
Clear character: ','
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

REFERENCES_MOD_FORM (CONTINUED)

Form REFERENCES_MOD_FORM
Form Definition

22-OCT-1987 16:29:37 VAX-11 FDU V1.4
22-OCT-1987 16:29:37 SYS\$INPUT: [] .COM; (1)

Page 4

7,7 Field name: AUTHOR_2
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPCASE,
Display attributes: REVERSE,

7,43 Field name: AUTHOR_4
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPCASE,
Display attributes: REVERSE,

10,1 Field name: TITLE_1
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

11,1 Field name: TITLE_2
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
OF POOR QUALITY

Page 5

REFERENCES_MOD_FORM (CONTINUED)

22-OCT-1987 16:29:37 VAX-11 FDU V1.4
22-OCT-1987 16:29:37 SYS\$INPUT: [] .COM; (1)

Form REFERENCES_MOD_FORM
Form Definition

13,27 Field name: ORGANIZATION
Field length: 30
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

15,45 Field name: DOCUMENT_NO
Field length: 30
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

17,30 Field name: DATE
Field length: 11
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

19,18 Field name: CONTRACT_NO
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

REFERENCES_MOD_FORM (CONTINUED)

22-OCT-1987 16:29:37 VAX-11 FDU V1.4
 22-OCT-1987 16:29:37 SYS\$INPUT:[] .COM; (1)

Form REFERENCES_MOD_FORM
 Form Definition

21,51
 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1
 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

ORIGINAL PAGE IS
 OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_STO_FORM

Page 1

VAX-11 FDU V1.4
SYS\$INPUT: [] .COM; (1)

22-OCT-1987 16:29:50
22-OCT-1987 16:29:50

Form Definition
REFERENCES_STO_FORM

Form name: REFERENCES_STO_FORM
Form path name: _CDD\$TOP.FIPM.REFERENCES_STO_FORM
Help form name:
Help form path name: 1
Beginning line number: 23
Last line number: 80
Form screen width:
Date/time form was stored in CDD: 13-NOV-1986 09:46:22.23

Field Access Order List:
Field name Subscript

AUTHOR_1
AUTHOR_2
AUTHOR_3
AUTHOR_4
TITLE_1
TITLE_2
ORGANIZATION
DOCUMENT_NO
DATE
CONTRACT_NO
CONTINUE

REFERENCES_STO_FORM (CONTINUED)

FORM IMAGE

1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890							
STORE REFERENCES							
AUTHORS (LAST NAME, INITIALS):							
1) XXXXXXXXXXXXXXXXXXXXXXXX 3) XXXXXXXXXXXXXXXXXXXXXXXX							
2) XXXXXXXXXXXXXXXXXXXXXXXX 4) XXXXXXXXXXXXXXXXXXXXXXXX							
REPORT OR DOCUMENT TITLE:							
XX							
XX							
XX							
ORIGINATING ORGANIZATION: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX							
ORIGINATING ORGANIZATION'S DOCUMENT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXX							
DOCUMENT DATE (DD-MMM-YYYY): 99-AAA-9999							
CONTRACT NUMBER: XXXXXXXXXXXXXXXXXXXXXXXX							
CONTINUE (Y, N OR A): A							
MESSAGE:							
XX							
123456789012345678901234567890123456789012345678901234567890							

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES_STO_FORM (CONTINUED)

Form Definition REFERENCES_STO_FORM VAX-11 FDU V1.4 Page 3
22-OCT-1987 16:29:50 SYS\$INPUT:[]:COM; (1)
22-OCT-1987 16:29:50

FIELD DEFINITIONS

6,7 Field name: AUTHOR_1
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

6,43 Field name: AUTHOR_3
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,7 Field name: AUTHOR_2
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

7,43 Field name: AUTHOR_4
Field length: 25
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

REFERENCES_STO_FORM (CONTINUED)

Form REFERENCES_STO_FORM
 Form Definition
 22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT:[] .COM; (1)

Page 4

10,1
 Field name: TITLE_1
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

11,1
 Field name: TITLE_2
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

13,27
 Field name: ORGANIZATION
 Field length: 30
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

15,45
 Field name: DOCUMENT_NO
 Field length: 30
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

REFERENCES_STO_FORM (CONTINUED)

Page 5

22-OCT-1987 16:29:50 VAX-11 FDU V1.4
22-OCT-1987 16:29:50 SYSINPUT: [].COM; (1)

Form REFERENCES_STO_FORM
Form Definition

17,30 Field name: DATE
Field length: 9
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

19,18 Field name: CONTRACT_NO
Field length: 20
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: UPPERCASE,
Display attributes: REVERSE,

21,51 Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Display attributes: REVERSE, DISPLAY ONLY,

H-95

ORIGINAL PAGE IS
OF POOR QUALITY

H-96

(This page intentionally blank)

SYSTEMS_FIN_FORM

Page 1

22-OCT-1987 16:31:02 VAX-11 FDU V1.4
22-OCT-1987 16:31:02 SYS\$INPUT: [] .CDM; (1)Form Definition
SYSTEMS_FIN_FORMForm name: SYSTEMS_FIN_FORM
Form path name: _CDD\$TOP.FIPM.SYSTEMS_FIN_FORM

Help form name:

Help form path name:

Beginning line number: 1

Last line number: 23

Form screen width: 80

Date/time form was stored in CDD: 12-JAN-1987 10:08:33.17

Field Access Order List:

Field name Subscript

SYSTEM
SYSTEM_NAME

ITEM_1

ITEM_2

ITEM_3

ITEM_4

ITEM_5

ITEM_6

ITEM_7

ITEM_8

ITEM_9

ITEM_10

ITEM_11

ITEM_12

ITEM_13

ITEM_14

ITEM_15

REF_1

REF_2

REF_3

REF_4

REF_5

REF_6

REF_7

REF_8

REF_9

REF_10

CONTINUE

PRECEDING PAGE BLANK NOT FILMED

SYSTEMS_FIN_FORM (CONTINUED)

Page 2

VAX-11 FDU V1.4
SYS\$INPUT: [].COM; (1)

22-OCT-1987 16:31:02
22-OCT-1987 16:31:02

Form Definition
SYSTEMS_FIN_FORM

FORM IMAGE

1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890							
FIND SYSTEMS							
1	2	3	4	5	6	7	8
SYSTEM: XXXX							
SYSTEM NAME:							
XX							
CONSTITUENT ROCKETDYNE FMEA ITEMS:							
1) XXXX	4) XXXX	7) XXXX	10) XXXX	13) XXXX			
2) XXXX	5) XXXX	8) XXXX	11) XXXX	14) XXXX			
3) XXXX	6) XXXX	9) XXXX	12) XXXX	15) XXXX			
REFERENCE DOCUMENTS:							
1) XXXXX	3) XXXXX	5) XXXXX	7) XXXXX	9) XXXXX			
2) XXXXX	4) XXXXX	6) XXXXX	8) XXXXX	10) XXXXX			
CONTINUE (Y OR N): A							
MESSAGE:							
XX							
123456789012345678901234567890123456789012345678901234567890							

SYSTEMS_FIN_FORM (CONTINUED)

Form Definition SYSTEMS_FIN_FORM
 Form Definition
 22-OCT-1987 16:31:02 VAX-11 FDU V1.4
 22-OCT-1987 16:31:02 SYS\$INPUT: [] .COM; (1)
 Page 3

FIELD DEFINITIONS

4,9 Field name: SYSTEM
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

7,1 Field name: SYSTEM_NAME
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

11,7 Field name: ITEM_1
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

11,22 Field name: ITEM_4
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Page 4

22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT: [] .COM; (1)

Form Definition SYSTEMS_FIN_FORM

11,37 Field name: ITEM_7
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

11,53 Field name: ITEM_10
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

11,69 Field name: ITEM_13
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

12,7 Field name: ITEM_2
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Form Definition SYSTEMS_FIN_FORM
 22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT: [] .COM; (1)
 Page 5

12,22 Field name: ITEM_5
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

12,37 Field name: ITEM_8
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

12,53 Field name: ITEM_11
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

12,69 Field name: ITEM_14
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Page 6

22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT: [].COM; (1)

Form Definition
 SYSTEMS_FIN_FORM

13,7 Field name: ITEM_3
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

13,22 Field name: ITEM_6
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

13,37 Field name: ITEM_9
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

13,53 Field name: ITEM_12
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Form Definition SYSTEMS_FIN_FORM
 22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT:[] .COM; (1) Page 7

13,69 Field name: ITEM_15
 Field length: 4
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

17,7 Field name: REF_1
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

17,22 Field name: REF_3
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

17,37 Field name: REF_5
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Page 8

22-OCT-1987 16:29:50 VAX-11 FDU V1.4
22-OCT-1987 16:29:50 SYS\$INPUT:[] .COM; (1)Form Definition
SYSTEMS_FIN_FORM

17,52 Field name: REF_7
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

17,68 Field name: REF_9
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

18,7 Field name: REF_2
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

18,22 Field name: REF_4
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Form Definition SYSTEMS_FIN_FORM
 22-OCT-1987 16:29:50 VAX-11 FDU V1.4
 22-OCT-1987 16:29:50 SYS\$INPUT: [].COM; (1) Page 9

18,37 Field name: REF_6
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

18,52 Field name: REF_8
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

18,68 Field name: REF_10
 Field length: 5
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, UPPERCASE,
 Display attributes: REVERSE,

21,48 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_FIN_FORM (CONTINUED)

Page 10

22-OCT-1987 16:29:50 VAX-11 FDU V1.4
22-OCT-1987 16:29:50 SYS\$INPUT:[] .COM; (1)

Form SYSTEMS_FIN_FORM
Form Definition

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,

SYSTEMS_MOD_FORM

Form Definition SYSTEMS_MOD_FORM 22-OCT-1987 18:31:17 VAX-11 FDU V1.4 Page 1
 22-OCT-1987 18:31:17 SYS\$INPUT: [] :COM; (1)

Form name: SYSTEMS_MOD_FORM
 Form path name: _CDD\$TOP.FIPM.SYSTEMS_MOD_FORM
 Help form name:
 Help form path name:
 Beginning line number: 1
 Last line number: 23
 Form screen width: 80
 Date/time form was stored in CDD: 12-JAN-1987 15:51:12.41

Field Access Order List:
 Field name Subscript

SYSTEM_NAME

ITEM_1
 ITEM_2
 ITEM_3
 ITEM_4
 ITEM_5
 ITEM_6
 ITEM_7
 ITEM_8
 ITEM_9
 ITEM_10
 ITEM_11
 ITEM_12
 ITEM_13
 ITEM_14
 ITEM_15
 REF_1
 REF_2
 REF_3
 REF_4
 REF_5
 REF_6
 REF_7
 REF_8
 REF_9
 REF_10
 CONTINUE

SYSTEMS_MOD_FORM (CONTINUED)

Page 2

VAX-11 FDU V1.4
SYS\$INPUT:[].COM; (1)

Form	SYSTEMS_MOD_FORM
Form Definition	

FORM IMAGE

	1	2	3	4	5	6	7	8
123456789012345678901234567890123456789012345678901234567890								
				MODIFY SYSTEMS				
				RECORD 99999 OF 99999				
				SYSTEM: A999				
				SYSTEM NAME:				
				XXX				
				CONSTITUENT ROCKETDYNE FMEA ITEMS:				
				1) A999	4) A999	7) A999	10) A999	13) A999
				2) A999	5) A999	8) A999	11) A999	14) A999
				3) A999	6) A999	9) A999	12) A999	15) A999
				REFERENCE DOCUMENTS:				
				1) AA999	3) AA999	5) AA999	7) AA999	9) AA999
				2) AA999	4) AA999	6) AA999	8) AA999	10) AA999
				CONTINUE (Y, N OR A): A				
				MESSAGE:				
				XXX				
123456789012345678901234567890123456789012345678901234567890								

CONTINUE (Y, N OR A): A

MESSAGE:

XXXXXXXXXXXXXXXXXXXXX
12345678901234567890123456789012345678901234567890

SYSTEMS_MOD_FORM (CONTINUED)

Form Definition SYSTEMS_MOD_FORM
 22-OCT-1987 16:31:17 VAX-11 FDU V1.4
 22-OCT-1987 16:31:17 SYSINPUT: [] :CDM; (1)
 Page 3

FIELD DEFINITIONS

3,37 Field name: RECORD_NUMBER
 Field length: 5
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: '0'
 Clear character: ','
 Display attributes: REVERSE, DISPLAY ONLY,

3,46 Field name: TOTAL_RECORDS
 Field length: 5
 Field scale factor: 0
 Field picture type: UNSIGNED NUMERIC
 Field datatype: UNSIGNED NUMERIC
 Fill character: '0'
 Clear character: ','
 Display attributes: REVERSE, DISPLAY ONLY,

5,9 Field name: SYSTEM
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Display attributes: REVERSE, DISPLAY ONLY,

8,1 Field name: SYSTEM_NAME
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: ','
 Clear character: ','
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Page 4

22-OCT-1987 16:31:17 VAX-11 FDU V1.4
 22-OCT-1987 16:31:17 SYS\$INPUT: [] .COM; (1)

Form SYSTEMS_MOD_FORM
 Form Definition

12,7 Field name: ITEM_1
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

12,22 Field name: ITEM_4
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

12,37 Field name: ITEM_7
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

12,53 Field name: ITEM_10
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Form	SYSTEMS_MOD_FORM	22-OCT-1987 18:31:17	VAX-11 FDU VI.4	Page 5
Form Definition		22-OCT-1987 18:31:17	SY\$INPUT: [].COM; (1)	

12,69 Field name: ITEM_13
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,,
 Clear character: ,,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,7 Field name: ITEM_2
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,,
 Clear character: ,,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,22 Field name: ITEM_5
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,,
 Clear character: ,,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,37 Field name: ITEM_8
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,,
 Clear character: ,,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Page 6

22-OCT-1987 16:31:17 VAX-11 FDU V1.4
 22-OCT-1987 16:31:17 SYS\$INPUT: [] .COM; (1)

Form Definition
 Form SYSTEMS_MOD_FORM

13,53
 Field name: ITEM_11
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,69
 Field name: ITEM_14
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

14,7
 Field name: ITEM_3
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

14,22
 Field name: ITEM_6
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: ,
 Clear character: ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Form	SYSTEMS_MOD_FORM	22-OCT-1987 16:31:17	VAX-11 FDU V1.4	Page 7
Form Definition		22-OCT-1987 16:31:17	SYSSINPUT: [:CDM; (1)	

14,37 Field name: ITEM_9
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

14,53 Field name: ITEM_12
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

14,69 Field name: ITEM_15
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

16,7 Field name: REF_1
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Form SYSTEMS_MOD_FORM
Form Definition

22-OCT-1987 18:31:17 VAX-11 FDU V1.4
22-OCT-1987 18:31:17 SYSSINPUT:[] .COM; (1)

Page 8

18,22 Field name: REF_3
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

18,37 Field name: REF_5
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

18,52 Field name: REF_7
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

18,68 Field name: REF_9
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Form	SYSTEMS_MOD_FORM	22-OCT-1987 16:31:17	VAX-11 FDU V1.4	Page 9
Form Definition		22-OCT-1987 16:31:17	SYS\$INPUT: [].COM; (1)	

19,7 Field name: REF_2
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

19,22 Field name: REF_4
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

19,37 Field name: REF_6
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

19,52 Field name: REF_8
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_MOD_FORM (CONTINUED)

Form Definition SYSTEMS_MOD_FORM 22-OCT-1987 16:31:17 VAX-11 FDU V1.4 Page 10
 Form Definition 22-OCT-1987 16:31:17 SYS\$INPUT: [] .COM; (1)

19,88 Field name: REF_10
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

21,51 Field name: CONTINUE
 Field length: 1
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

23,1 Field name: MESSAGE
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Display attributes: REVERSE, DISPLAY ONLY,

SYSTEMS_STO_FORM

Form Definition
 Form name: SYSTEMS_STO_FORM
 Form path name: _CDD\$TOP.FIPM.SYSTEMS_STO_FORM
 Help form name:
 Help form path name:
 Beginning line number: 1
 Last line number: 23
 Form screen width: 80
 Date/time form was stored in CDD: 13-NOV-1986 09:46:42.52

22-OCT-1987 16:31:32 VAX-11 FDU V1.4
 22-OCT-1987 16:31:32 SYS\$INPUT: [].COM; (1)

Page 1

Field Access Order List:
 Field name Subscript

SYSTEM
 SYSTEM_NAME
 ITEM_1
 ITEM_2
 ITEM_3
 ITEM_4
 ITEM_5
 ITEM_6
 ITEM_7
 ITEM_8
 ITEM_9
 ITEM_10
 ITEM_11
 ITEM_12
 ITEM_13
 ITEM_14
 ITEM_15
 REF_1
 REF_2
 REF_3
 REF_4
 REF_5
 REF_6
 REF_7
 REF_8
 REF_9
 REF_10
 CONTINUE

SYSTEMS_STO_FORM (CONTINUED)

Page 2

VAX-11 FDU V1.4
SYS\$INPUT: [].COM; (1)Form Definition
SYSTEMS_STO_FORM

FORM IMAGE

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890							
STORE SYSTEMS							
1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	
3	4	5	6	7	8		
4	5	6	7	8			
5	6	7	8				
6	7	8					
7	8						
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
SYSTEM NAME:							
XX							
CONSTITUENT ROCKETDYNE FMEA ITEMS:							
1) A999 4) A999 7) A999 10) A999 13) A999							
2) A999 5) A999 8) A999 11) A999 14) A999							
3) A999 6) A999 9) A999 12) A999 15) A999							
REFERENCE DOCUMENTS:							
1) AA999 3) AA999 5) AA999 7) AA999 9) AA999							
2) AA999 4) AA999 6) AA999 8) AA999 10) AA999							
MESSAGE:							
XX							
12345678901234567890123456789012345678901234567890							
1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	
3	4	5	6	7	8		
4	5	6	7	8			
5	6	7	8				
6	7	8					
7	8						
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
CONTINUE (Y, N OR A): A							

SYSTEMS_STO_FORM (CONTINUED)

Form Definition SYSTEMS_STO_FORM
 22-OCT-1987 18:31:32 VAX-11 FDU V1.4
 22-OCT-1987 18:31:32 SYS\$INPUT: [] : CDM; (1) Page 3

FIELD DEFINITIONS

4,9 Field name: SYSTEM
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

7,1 Field name: SYSTEM_NAME
 Field length: 80
 Field picture type: ALPHANUMERIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

11,7 Field name: ITEM_1
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

11,22 Field name: ITEM_4
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Page 4

22-OCT-1987 16:31:32 VAX-11 FDU V1.4
 22-OCT-1987 16:31:32 SYS\$INPUT: [] .COM; (1)

Form Definition SYSTEMS_STO_FORM

```

11,37 Field name: ITEM_7
      Field length: 4
      Field picture type: ALPHABETIC
      Field datatype: TEXT
      Fill character: ,
      Clear character: ,
      Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
      Display attributes: REVERSE,

11,53 Field name: ITEM_10
      Field length: 4
      Field picture type: ALPHABETIC
      Field datatype: TEXT
      Fill character: ,
      Clear character: ,
      Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
      Display attributes: REVERSE,

11,69 Field name: ITEM_13
      Field length: 4
      Field picture type: ALPHABETIC
      Field datatype: TEXT
      Fill character: ,
      Clear character: ,
      Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
      Display attributes: REVERSE,

12,7  Field name: ITEM_2
      Field length: 4
      Field picture type: ALPHABETIC
      Field datatype: TEXT
      Fill character: ,
      Clear character: ,
      Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
      Display attributes: REVERSE,

```

SYSTEMS_STO_FORM (CONTINUED)

Form	SYSTEMS_STO_FORM	22-OCT-1987 16:31:32	VAX-11 FDU V1.4	Page 5
Form Definition		22-OCT-1987 16:31:32	SYS\$INPUT: [].COM; (1)	

12,22
Field name: ITEM_5
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

12,37
Field name: ITEM_8
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

12,53
Field name: ITEM_11
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

12,69
Field name: ITEM_14
Field length: 4
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: ,
Clear character: ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Form Definition SYSTEMS_STO_FORM
 22-OCT-1987 16:31:32 VAX-11 FDU V1.4
 22-OCT-1987 16:31:32 SYS\$INPUT: [].COM; (1) Page 6

13,7 Field name: ITEM_3
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,22 Field name: ITEM_8
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,37 Field name: ITEM_9
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

13,53 Field name: ITEM_12
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Form SYSTEMS_STO_FORM 22-OCT-1987 16:31:32 VAX-11 FDU V1.4 Page 7
 Form Definition 22-OCT-1987 16:31:32 SYS\$INPUT: [] .COM; (1)

13,69 Field name: ITEM_15
 Field length: 4
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

17,7 Field name: REF_1
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, RESPONSE REQUIRED, UPPERCASE,
 Display attributes: REVERSE,

17,22 Field name: REF_3
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

17,37 Field name: REF_5
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Form Definition SYSTEMS_STO_FORM
 22-OCT-1987 16:31:32 VAX-11 FDU V1.4
 22-OCT-1987 16:31:32 SYS\$INPUT: [] .COM; (1) Page 8

17,52 Field name: REF_7
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

17,68 Field name: REF_9
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

18,7 Field name: REF_2
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

18,22 Field name: REF_4
 Field length: 5
 Field picture type: ALPHABETIC
 Field datatype: TEXT
 Fill character: , ,
 Clear character: , ,
 Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
 Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Form Definition SYSTEMS_STO_FORM Page 9

22-OCT-1987 16:31:32 VAX-11 FDU V1.4
22-OCT-1987 16:31:32 SYS\$INPUT: [].COM; (1)

18,37 Field name: REF_6
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

18,52 Field name: REF_8
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

18,88 Field name: REF_10
Field length: 5
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: AUTOTAB, MUST FILL, UPPERCASE,
Display attributes: REVERSE,

21,51 Field name: CONTINUE
Field length: 1
Field picture type: ALPHABETIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Attributes assigned: MUST FILL, RESPONSE REQUIRED, UPPERCASE,
Display attributes: REVERSE,

SYSTEMS_STO_FORM (CONTINUED)

Form	SYSTEMS_STO_FORM	22-OCT-1987 16:31:32	VAX-11 FDU V1.4	Page 10
Form Definition		22-OCT-1987 16:31:32	SY\$INPUT:[]:COM; (1)	

23,1 Field name: MESSAGE
Field length: 80
Field picture type: ALPHANUMERIC
Field datatype: TEXT
Fill character: , ,
Clear character: , ,
Display attributes: REVERSE, DISPLAY ONLY,